DOCUMENT RESUME

ED 125 658 IR 003 750

TITLE Instructional (I). Papers Presented at the Association for Educational Data Systems Annual Convention (Phoenix, Arizona, May 3-7, 1976).

INSTITUTION Association for Educational Data Systems, Washington, D.C.

PUB DATE May 76

NOTE 93p.; For related documents, see IR 003 748-756; Some parts may be marginally legible due to print quality of original

AVAILABLE FROM Association for Educational Data Systems, 1201 Sixteenth Street, N.W., Washington, D.C. 20036 ($10.00 for entire proceedings)

EDRS PRICE MF-$0.83 HC-$4.67 Plus Postage.

DESCRIPTORS Business Education; Career Planning; *Computer Assisted Instruction; *Computer Science Education; *Data Processing Occupations; Diacritical Marking; Economic Education; Educational Innovation; Electronic Data Processing; Higher Education; Individualized Instruction; Information Science; Information Systems; Instruction; Laboratory Training; Mathematical Models; On Line Systems; Programers; *Simulation; Statistical Analysis; Test Construction

IDENTIFIERS AEDS 76; *Association for Educational Data Systems; Interactive Computer Systems; Programable Calculators

ABSTRACT

Sixteen articles and two abstracts on the use of computers and electronic equipment in instruction presented at the Association for Educational Data Systems (AEDS) 1976 convention are included here. Uses of the computer to generate and solve mathematical models, to generate examinations, and to facilitate concept learning are examined. Six articles discuss simulation in the following areas: biology, macroeconomics, management, and business policy. Papers on computer assisted instruction cover instruction in accounting, geography, programing, diacritical marking, economics, and laboratory methods. One paper examines prediction of success in data processing training, and another discusses the use of programable, hand-held calculators for calculus instruction. A suggested programing curriculum for a small college is outlined. (CH)

****
* Documents acquired by ERIC include many informal unpublished *
* materials not available from other sources. ERIC makes every effort *
* to obtain the best copy available. Nevertheless, items of marginal *
* reproducibility are often encountered and this affects the quality *
* of the microfiche and hardcopy reproductions ERIC makes available *
* via the ERIC Document Reproduction Service (EDRS). EDRS is not *
* responsible for the quality of the original document. Reproductions *
* supplied by EDRS are the best that can be made from the original. *
****

ED125658

INSTRUCTIONAL (I). Papers presented at the Association for Educational Data Systems Annual Convention.

Benn Konsynski
Peter Braun
Jeff Schnepper
Frank Aldrich
Kathleen A. Powers
James A. Robb
Paul Cheney
James R. von Feldt
Joanne Subtelny
Frank L. Kaufman
B. M. Hall
F. F. Liu
William B. Wood
Richard M. Hogetts
William W. Wells
Martin T. Lang
William S. Dorn
Roy Elliot
Dale A. Vorderlandwehr
J. J. Lagowski

BEST COPY AVAILABLE

U S DEPARTMENT OF HEALTH, EDUCATION & WELFARE
NATIONAL INSTITUTE OF EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

IR 003 735-O

BEST COPY AVAILABLE

ASSOCIATION FOR EDUCATIONAL DATA SYSTEMS

The 1976 AEDS International Convention

Proceedings

TODAY'S REVOLUTION: COMPUTERS IN EDUCATION

May 3 - 7, 1976

Phoenix, Arizona

These Proceedings have been made possible by a grant from the IBM Corporation as a service to the educational community.

A PEDAGOGICAL TOOL FOR TRAINING SYSTEMS ANALYSTS

Benn R Konsynski
Management Information Systems
Business and Public Administration
University of Arizona
Tucson, Arizona 85721

A model of a teaching aid and a description of its usage is presented. The system is used to make students aware of the problems involved in collection and organization of information in an information system design effort. A data base of case facts concerning specification of requirements in design of an information system along with a limited query facility are utilized to simulate some of the behavioral and technical problems the analyst will encounter.

INFORMATION SYSTEMS AND ORGANIZATIONS

Information systems exist in organizations to monitor and regulate operations. The systems consist of the data and manipulators that collect, process and distribute information to the operating components of the system. Computer-based information systems utilize the mass and high speed capabilities in data storage and processing that characterize computers. In further disussions we will refer to these computer-based systems.

Information systems are a complex of men, material and machines that create and manipulate information that is necessary and useful in the operation of an organization. The complexity of the interactions and the variety of necessary system interfaces serve to make the analyst's task a complex undertaking. Simple documentation of a system is often impossible due to the degree of effort necessary to effectively document the requirements and the volatility and complex im[illegible] of change.

Nevertheless, computer based information systems are not only here to stay, but the demand for qualified analysts is ever increasing. To meet this need, the education process for future analysts must be formalized. The analysts must be trained not only to serve industry's present needs, but must also be prepared to enter a field whose growth in the next 10-20 years will match or surpass that of computer hardware in the past decade. This situation is the result of the tremendous demand for application systems and the ever rising costs and complexity in their development.

THE ANALYST AND THE DESIGN PROCESS

As information systems are complex, so to is the design process. The process of carrying a system from the perception of need to its implementation is a highly under-rated task, as many organizations have painfully learned over and over again. The reliance on untrained analysts using only ad hoc techniques has proven both costly and operationally disasterous. The ever increasing sophistication of hardware and software as well as rising demand for responsible and secure systems increase our concern.

The design process consists of four basic steps:

1. Definition of user requirements.
2. Logical and Physical system design.
3. Construction and implementation.
4. Operation and evaluation.

The first stage includes user recognition of the need for the system, an establishment of the user requirements and feasibility of operation and interface, detailed definition and documentation of the user requirements, and analysis of those requirements for completeness and consistency. This stage usually involves a significant degree of communication between the user and the analyst in deriving specifications of user needs. There is little formal methodology in this area and the effectiveness of this stage is often a function of the analyst's capabilities in interpreting user needs and gathering and organizing sufficient information to proceed.

The second stage involves the determination of data organization, specification of program modules, and selection of necessary hardware. The analyst must analyze the previously defined requirements under the constraints of operations and hardware availability. Success in this stage is often a function of the analysts technical skill ability to grasp all relevant aspects of the problem.

The third stage of the design process consists of code generation, file and data base creation, and implementation and testing of the system prior to release for operation. In this stage the analyst and programmer work in concert to create the necessary code and define data structures for the various programs. The analyst must direct the programmer's implementation to insure that the system will be true to the user specifications.

The final stage includes operation, monitoring, and performance evaluation in terms of both utility to the user and effeciency of resource utilization. Again the analyst plays a key role by analyzing user feedback and systems staff resource usage reports in the context of the system requirements and design specifications. This stage often leads to a determination of a need for modification of user requirements or aspects of the implementation.

Historically, the function of the programmer has been handled by the analyst on the basis of economics. However, with the increased complexity and scope of systems efforts, the division of labor is becomming increasingly apparent.

EDUCATION AND TRAINING OF ANALYSTS

As stated earlier, analysts characteristically have little formal training for their role. Most analysts evolve from a programming background, so most of the training is done on the job.

It is considered too expensive to allow future analysts to gain experience in managing real-world projects. Indeed, as little formal methodology is used in practice, the analysts develops his, or her, good or bad habits through experience alone. Thus, the analyst learns by participating at lower levels and gradually gains more and more responsibility.

At present, several groups are developing curricula to provide the basic background necessary in analysis. For instance, the ACM Curriculum Committee on Computer Education for Management see Ashenhurst, 1972, and Couger, 1973, have made recommendations on technical and organizational skills the analyst would find useful. It is interesting to note that both industry and academia were involved in this effort.

A LABORATORY OF ANALYST TRAINING

The primary tool that needs to be developed is a laboratory for training analysts. The laboratory would consist of a specially designed case study and necssary material for performing systems design according to alternative methodologies. Such a facility would provide an environment for the beginning analyst to explore differing methodologies and compare alternative decisions. It would also provide a means of monitoring and evaluating the analysts progress.

The case study must be designed according to some of the following specifications:

- The case must represent typical real-world problems.
- The complexity must be representitive, yet the effort involved in the design must not be too excessive as to make the design effort the end rather than the learning experience.
- The case must be logically complete so that sufficient information is available for the design process.
- One must be able to apply many of the developing methodologies to the case design.
- The case must be modifiable or extendable in order to evaluate the impact of change.

A laboratory would allow the student analyst to formalize the duties of the analyst and evaluate alternative procedures. The analyst would be able to experience the differing roles of the user, analyst and programmer and in this way identify responsibilities.

The student would gain experience with differing techniques and be able to evaluate the work of others in the context of the design that the student arrives at. Further, the experience of differing team organizations and role playing would be a major part of the experience.

The laboratory can prove to be a center for evaluation of alternative tools. The benifit of and conditions under which certain tools prove useful could be determined. The utility of such diverse

tools as simulation, data dictionaries, program development monitors, PERT/CPM, and decision table processors might be considered.

The evaluation of proposed methodologies can be explored. Methodologies such as those proposed by ARDI, ISDOS, SOP, and HOSKINS could be compared and appropriate conclusins drawn for modification or evolution of a new approach. The laboratory would provide a basis of comparison as well as a means of communicating as the different approaches have differing formal definitions of terms used.

The laboratory environment would provide valuable experience for real-world design environment. Experienced, up-to-date analysts would be produced and the state of the art advanced. In addition, the progress of each student could be evaluated.

THE ANALYSIS RESPONSIBILITIES

The analyst is the key individual in a system design effort. It is the analyst that initially guides the user in determining the details of the requirements. It is the analysts task to define, in detail, the requirements and see to it that all the necessary information is specified. The analyst conducts the design process and works with the programmer and data base administrator in constructing the system. Finally, it is usually the analyst that conducts the evaluation and guides the necessary modifications.

Once a user, or user group, determines that there is a need for an information system, the analyst's task begins. The user must gain some insight in regard to the feasibility of the system as well as the feasibility of the system effort. The analyst must crudely estimate the investment in time, manpower and money necessary to realize the system to satisfy the stated user requirements. While little information is available for the analyst to work with, the analyst must provide responsible answers to feasibility questions.

The analyst must aid the user in determination of the system requirements. The analyst first must define the essential outputs, then the processing necessary to generate those outputs. Next, the user and the analyst must determine the necessary and available inputs through analysis of existing and planned source documents. For those necessary inputs that are not presently available, the analyst must design forms and determine procedures for introducing that data gathering function into the organization.

In order to perform satisfactory design processes, the analyst must document the requirements in detail. This has, till recently, accomplished only in a narrative form. Several languages have been developed to document system requirements, see Teichroew, 1974. The future analyst must be made familar with the important language and requirements definition techniques.

The analyst must be trained to analyze the documented requirements. The requirements must be analyzed for completeness of definition and consistency. While computer-aids are being developed for this analysis, see Munamaker and Konsysski, 1975, the student should develop the analysis rules and procedures through manual analysis of the case.

The design process is one of the weakest areas in analysis texts. The design process is pictured as a creative process in which skill can be developed only over time. While design is still basically a manual task, the student can be taught many of the developing models and computer-aided techniques.

With respect to data organization, the analyst should be introduced to concepts of data representation, record and file design and logical data organization. With the increasing importance of data management systems, considerable effort should go into problems of data base design and creation, data administration and application program interfaces.

The future analyst must posess a basic understanding of software engineering principles and structured design approaches. The analyst must deal with the systems of the future, with multiple processors, segmented memories and distributed data. The faults of functional decomposition in determination of program modules must be demonstrated.

The existing procedures for hardware selection must be taught. The situations in which scoring, benchmarks, kernels and other selection techniques, are applicable should be part of the training. While not requiring the student to have hardware skills, the student must understand the basics of configuring a system.

As stated earlier, the analyst must formulate his design decisions for programmers to implement. This communication with the programming staff must be precise and overviewed. The analyst must see that the design specifications are correctly implemented.

A COURSE IN SYSTEMS ANALYSIS AND DESIGN

A course in systems analysis must prepare the analyst for all of the above responsibilites. The course must stress those operational problems the analyst will encounter. The course should follow the basic life cycle of systems as outlined discussions on the design process.

To this end, a project oriented course seems most likely to provide the basic skills while providing valuable experience. This includes basic classroom instruction on basic methodologies and available tools, while allowing application of these techniques in a realistic situation. The example used is often either an actual company system effort or an imaginary one presented as a case study.

Although a real design situation might seem to be an ideal training laboratory, the environment of a functioning organization is not a very good place to begin. Aside form any disruptive effect that training activities may have upon the organization or any ongoing design effort, the real problems lack the control necessary to insure that all aspect are covered with sufficient significance. While the handling of the mass of detail is important, the training analyst should not get bogged down in detail.

In order to maintain control of the problem,in order that the student will acquire the necessary concepts and principles, case studies are used as control problems representing actual design situations. Case studies are adequate for teaching many aspects of the design process.

A TOOL FOR ANALYST TRAINING

Whether intended for the teaching of industrial management principles or information systems analysis and design, case studies do not develop one of the fundamental skills, namely the collection of necessay data for solution of the problem. The student must develop the ability to collect data in a realistic manner in order to asses incompleteness and inconsistency situations. Case studies too often present the facts in a way that misleads the student as to the availability of correct and necessary data.

To this end, an information systems design tool is being adapted for use in providing the student analyst with relevant experience in performing a system study. The aid provides for simulation of the environment encountered during the analysis and design of a complex information system.

The system consists of a data base and a query facility. The tool simulates the dialogue that occurs between the analyst and the user in determining and documenting system requirements. The data base contains the requirements for a hypothetical company (Company Z), see Nunamaker and Konsynski, 1974). The query facility allows the student to question the requirements in building a documentation of the requirements in another form (ADS, TAG or some other system language).

The student is made aware of the complex data collection and handling problems. In this way, the student learns:

What questions to ask? What answers demand what questions? Who to ask questions of?

SYSTEMS DESIGN TOOL

The vehicle for implementation of the requirements data base and its query facility is the Problem Statement Analyzer (PSA) developed by the Information Systems Design and Optimization System (ISDOS) Project, see Nunamaker and Konsynski, 1975. PSA is a software package that maintains a data base of requirements that have been defined in the Problem Statement Language (PSL), a systems requirement statement language. The Company Z example was described in PSL and a PSA data base produced.

The query and reporting capabilities of the PSA system were limited as to simulate the necessarily limited response provided by humans. Thus, armed with this limited subset of the PSA commands, the student proceeds to gather and analyze the requirements.

HIGHLIGHTS OF THE PROJECT

As space does not permit a description of the PSL/PSA system, see Nunamaker and Konsynski, 1975, some highlights of the project might be addressed:

The Company Z case study has developed over the past five years and has been used by the author and other teachers as a design example. It represents a medium size manufacturer-wholesaler of electrical parts.The sales for the example run twelve million dollars. The case has been designed to represent many of the important design decisions. The case is representative of basic complexity encountered in real situations, yet does so without excessive and unnecessary detail. For the present, the complete case is maintained in the data base.

The student proceeds by first determining the high-level structure of the organization. He is told to generate a Document and Material Flow Diagram. Next he must begin to develop a documentation of the requirements by successive queries. The student must then use this information to perform a system design and select and configure a hardware system.

The students work in teams and are given milestones and due dates for different reporting stages. Several times the class teams served in different functions or focused on different aspects of the case and were told to interface with other teams.

EXTENSIONS AND CONCLUSIONS

The tool provides the basic facility necessary for creation of a laboratory for information systems work. The extensions and modifications possible are limitless. Some of the major extensions might include:

The system is being developed as a tool in the Information System curriculum at the University of Arizona. It is expected that this tool will provide valuable experience in preparation for real-world design problems while facilitating the monitoring of student development.

Determination of the adequacy of the present form of queries. Thus providing for additions or deletions from the query capabilities.

Introduction of a monitoring capability. One could then count the questins by type and measure against the minimal necessary to acquire a complete description. The potential exists for evaluating the type and timing of questions with a monitoring facility.

Incompleteness and inconsistencies might be introduced into the data base inorder to evaluate the student's ability to recognise such situations.

The data base can be used to evaluate system design tools.

A feature might be introduced to facilitate random supression of information. This means not to lie, just dont tell the whole truth at one time. This would provide for simulation of the real-life situations of forgetfulness, resistence and undefined responsibilities.

It is expected that the application of such a pedagogical tool to education of future analysts will provide valuable experience in preparation for real-world design situations. In addition, information system educators will posses a useful facility for monitoring student development and conducting experiments.

BIBLIOGRAPHY

Ashenhurst, R. L., ed. Curriculum Recommendations for Graduate Professional Programs in Information Systems. Communications of the ACM, V15, N5, May, 1972.

Couger, J. D., ed. Curriculum Recommendations for Undergraduate Programs in Information Systems, Communications of the ACM, V16, N12, December, 1973.

Nunamaker, J. F. and Konsynski, B. R., Company Z: an Information System Design Problem, College of Business and Public Administration, University of Arizona, Tucson, December, 1974.

Nunamaker, J. F. and Konsynski, B. R., "From Problem Statement to Automatic Code Generation", SYSTEMEERING 75, Student Literatur, Lund, Sweden, 1975.

Teichroew, D., "Improvements in the System Life Cycle", IFIPS, 1974.

USING A COMPUTER TO FACILITATE CONCEPT LEARNING

Peter H. Braun
Institute for Behavioral Research
York University, Toronto

ABSTRACT: To use a computer in teaching has often been an experience involving additional work loads for students and teachers. Much effort was required for learning special languages comprehensible to a computer, and for translating instructions formulated in those languages onto special media such as punched cards or tape. Fortunately, recent innovations in both hardware and software have made available facilities where a student or teacher can communicate with a computer in much more natural fashion, using terms from his native language, and concepts developed since his early days in school. This paper illustrates how an advanced problem-solving language can be used to solve problems on the blackboard, and how solutions can be immediately verified on a computer without the need for translation or advanced computer knowledge.

INTRODUCTION

Recent psychological research has pointed the way to more effective teaching methods. One result is that the modern teacher aims to help his students reach a satisfactory level of understanding of his discipline; the emphasis in this new approach being on understanding as opposed to rote learning. The various methodological approaches used to teach a given subject in this way are usually called 'insight learning' or 'discovery learning'. Their goal is to encourage students to reach a high level of understanding of a given concept by illuminating the properties of the concept and its connection with related concepts. In terms of behavioral objectives, the teacher is aiming at the third level of behavior illustrated in Table 1.

TABLE 1

THREE LEVELS OF BEHAVIORS

LEVELS	*DESCRIPTION*	*EXAMPLES OF BEHAVIORS*	
1	*SIMPLE BEHAVIOR*	*GATHER DATA*	*MEASURE AND INVESTIGATE*
2	*MORE COMPLEX MENTAL OPERATIONS*	*ORGANIZE DATA*	*ANALYZE*
3	*BEHAVIOR SHOWING UNDERSTANDING*	*GENERALIZE DATA*	*MAKE INFERENCES*

Behavioral evidence of a student's understanding is defined as his ability to generalize appropriately from a familiar instance of a concept to a new instance. In other words, to say that a student has gained insight into a topic means that he has come to see its underlying structure, and that, by referring to this structure, he is able to state and apply rules that explain or predict phenomena associated with the topic. In particular, he is able to abstract the rules from their context, and generalize them to new situations. However, to do this requires symbolic thought; and in order to communicate their thinking, teachers and students need a language suitable for expressing scientific processes clearly and without ambiguity. Hence, one prerequisite of an educational process is that a language be available in which the discipline can be described; a language in which one can think about the concepts and rules.

The APL-language (A Programming Language) provides an excellent tool for describing a discipline, and makes it easier to think symbollically about its concepts and rules. The phrase 'open use of APL' designates a teaching approach that takes full advantage of the features of APL in teaching the concepts of the discipline.

The concepts in 'open use of APL' can perhaps best be explained by showing a sample classroom application. In this paper, an application from statistics will be shown and discussed. Other examples of 'open use of APL' can be found in Spence (1974) who applied it to electric circuit theory, or IBM (1975), where its use is illustrated with Physics material.

Historically, APL has been used in many places for teaching purposes. But the credit for formulating the specific teaching strategy called 'open use' goes to the University of Bari (CSATA), and the IBM Scientific Center in Bari, Italy, who jointly developed the method as outlined in this paper, and elsewhere (IBM, 1975).

AN EXAMPLE FROM STATISTICS

Let us consider a problem from statistics, such as the problem of describing the degree of simultaneous variation of two variables. The data under consideration would consist of pairs of measurements: for example, measurements of both the height and weight of each child in a group of schoolchildren.

The relationship between the two variables is called their 'correlation', and the resulting correlation coefficient 'r' is an index of the degree to which the two variables are related.

Various methods for calculating r exist, but the most frequently used one is as follows:

$$r_{xy} = \frac{n\Sigma X_i Y_i - (\Sigma X_i)(\Sigma Y_i)}{\sqrt{[n\Sigma X_i^2 - (\Sigma X_i)^2]\,[n\Sigma Y_i^2 - (\Sigma Y_i)^2]}}$$

This function is specifically intended to make it easy to calculate r; but in so doing, it totally obscures the nature of the coefficient.

However, working with APL makes it unnecessary to have functions that make calculations easier. The computer can evaluate students' exercises in fractions of a second anyway, regardless of the complexity or efficiency of the algorithm. Therefore, the teacher can concentrate on presenting the concept of correlation in its computationally easiest form:

$$r_{xy} = \frac{\Sigma z_x z_y}{n-1}$$

This makes it quite clear that r is simply the average crossproduct of the standardized variables, adjusted for losses in degrees of freedom. This function can be executed in exactly this form in APL, so that no calculating formula need ever be introduced to obscure the simple nature of the coefficient r.

To obtain a correlation coefficient on an APL terminal, a student would enter his data for the height and weight of, let us say, five children as follows:

```
HEIGHT←156  138  145  129  141
WEIGHT←48.3 40.1 42.5 41.2 43.1
```

Then, he would type

```
CORRELATE HEIGHT AND WEIGHT
```

to obtain his correlation coefficient r.

If we contrast these statements with the algebraic expressions above requiring the calculation of all crossproducts, and sums of squares, etc., it will be obvious that a student will find it much easier to interpret the meaning of the APL statements. In APL, all variables and functions have self-explanatory, meaningful names rather than plain *X* and *Y* as in the commonly used algebraic expressions. This in itself facilitates understanding, because it eliminates one level of translation requiring the student to read *HEIGHT* for *X*, and *WEIGHT* for *Y*.

The *AND* function used above to connect the two variables *HEIGHT* and *WEIGHT* is called a "housekeeping" function. It transforms the *N*-element data strings *HEIGHT* and

WEIGHT into a data matrix of *N* rows and 2 columns, very similar to the way we generally arrange tabular data in the form of a matrix on a sheet of paper.

```
      HEIGHT AND WEIGHT
156     48.3
138     40.1
145     42.5
129     41.2
141     43.1
```

The *AND* function will further permit statements combining more than two variables, as for example:

```
      ENGLISH AND MATH AND HISTORY AND IQ
```

In this fashion, a data matrix of any size can easily be assembled.

This data matrix is then passed as an argument to the function *CORRELATE*, which might have the following definition:

```
      ∇ CORRELATION←CORRELATE DATAMATRIX
[1]     S←STANDARDIZE DATAMATRIX
[2]     N←ROWS DATAMATRIX
[3]     CORRELATION←((⍉S)+.×S)÷(N-1)
      ∇
```

To the uninitiated, this might look a bit strange; but with a few sentences of explanation, it is easy to understand. Remember that the user of APL has only his terminal to converse with the computer, and that all communications with the computer are made via this terminal. Hence, there must be some way to differentiate between data such as *HEIGHT* and *WEIGHT*, and functions, such as *CORRELATE*. This is accomplished by use of the symbol '∇' (called 'del'): whatever is enclosed between two dels is treated as a function definition. Once defined, such a function can then be called as part of a statement to act on an argument, just as shown above where the function *CORRELATE* is called to act on the argument *HEIGHT AND WEIGHT*.

The first line of a function definition (unnumbered) contains the name of the function, any arguments to be used, and the name of the result. The following lines (numbered) develop the result. Referring to the *CORRELATE* function displayed above, in line [1], the columns of the data matrix are standardized by using an appropriate function called *STANDARDIZE*, and the standardized matrix is assigned to *S*. In line [2], a housekeeping function *ROWS* determines the number of rows of the data matrix, and this is assigned to *N*. In line [3], the cross-multiplication and addition implied by

$$\Sigma z_x z_y$$

are executed by the APL primitive function +.× which expresses that two operations are involved - addition and multiplication. In fact, the expression +.× calculates the familiar inner product of matrix algebra. Therefore, *S* must be transposed to conform with the rules of matrix multiplication. This is done by means of the APL primitive function ⍉. Finally, the result of this matrix multiplication is divided by *N*-1 and assigned to *CORRELATION*.

Those familiar with matrix algebra may have already guessed that this function not only works for the bivariate case illustrated here, but applies to a data matrix of any size. Hence, the transition from bivariate to multi-variate statistics, which in conventional notation is a teacher's nightmare, is done automatically in APL with no problems of notation to hinder understanding.

At this stage, students would evidently have to be familiar with the concept of 'standardizing' variables for the purpose of comparison. Even to the statistical novice, however, it is probably obvious that two measures such as height and weight (the one measured in centimeters, the other in kilograms) may have to be transformed to some sort of standard scale before they can be usefully compared or related to each other.

The *CORRELATE* function demonstrates how easily and quickly the components of a function can be examined for their individual contributions. An examination of intermediate results can illuminate the components of a function, and thereby greatly enhance the understanding of the total function.

Suppose, a student had forgotten exactly what the *STANDARDIZE* function does. He can examine the intermediate results of this function by typing

```
      STANDARDIZE HEIGHT AND WEIGHT
 1.437      1.748
¯0.3844    ¯0.5098
 0.3237    ¯0.3642
¯1.295     ¯0.7284
¯0.0809    ¯0.1457
```

The computer immediately calculates the answer, and prints it for the student.

The function definitions discussed so far include only those concepts necessary for understanding them, and no more. This is a salient feature of open use of APL, which provides for concepts to be ordered in a hierarchy. The hierarchy can easily be charted in diagrammatic form, as shown in Figure 1, where the two functions discussed above are placed in a hierarchy of statistical concepts.

When one considers that the concept of 'correlation coefficient' may require several lectures of conventional classroom

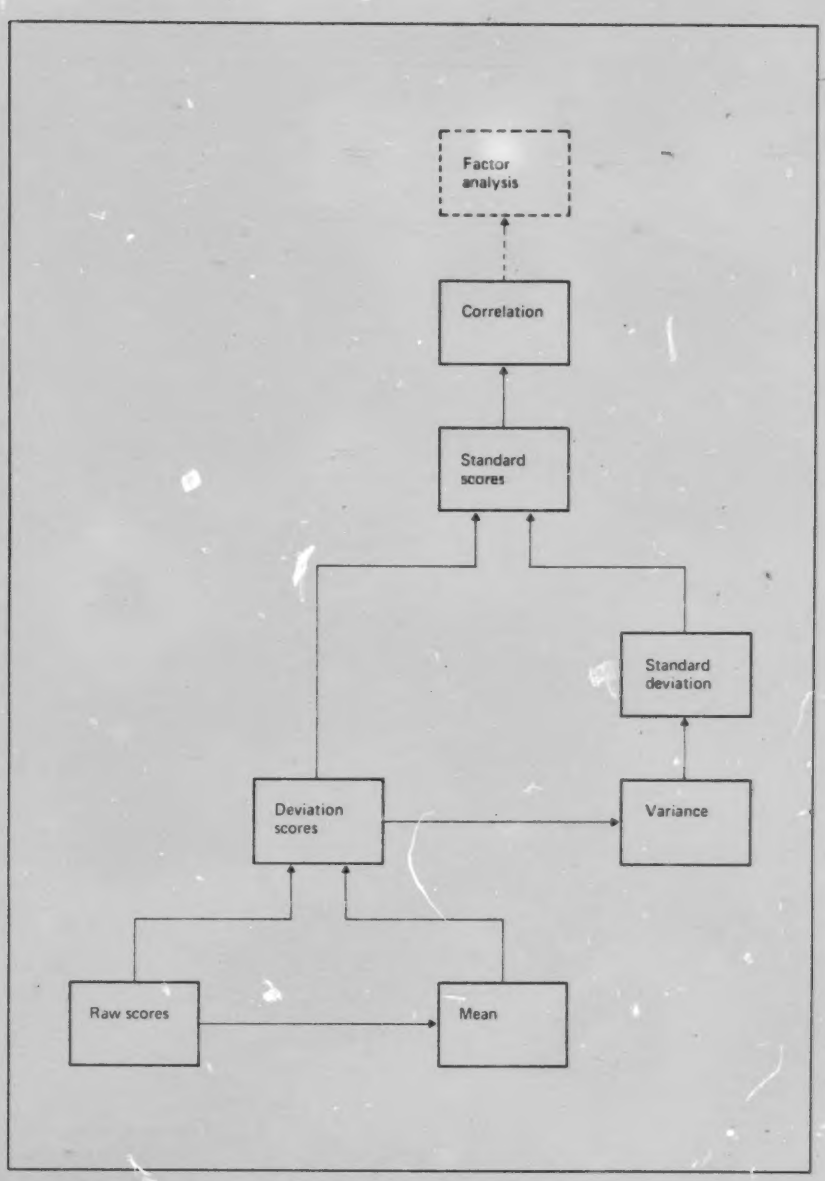


Figure 1. The hierarchy of statistical concepts required to understand *CORRELATION*

instruction in order that the calculating algorithms shown earlier may be explained and developed, during the course of which much time may be wasted and the students' interest lost, one can appreciate the simplicity and efficiency of open use of APL, which explains correlation by building on earlier concepts. This building block approach greatly improves understanding, since it does not obscure the process of learning with the mechanics of calculating the sums of squares and cross-products - a task much better delegated to the computer.

'Open use' of APL is a teaching technique that enhances 'insight' or 'discovery' learning. But it is more than a teaching technique: it enhances the process of insight learning with the help of the computer.

The computer, with its large memory, its fast access to stored data, and its calculating ability, gives this method of teaching a power and flexibility that are limited only by the imagination of the teacher and his students.

In the forgoing examples, APL has been employed in a deliberate effort to transcend the usual role of programming languages: APL not only simplifies the control of the computer, but also simplifies the statement of procedure. Thus, the APL expressions shown earlier can be used not only on the computer, but also, equally well, in books or on the blackboard to illustrate and explain the meaning of correlation. APL expressions can be used to achieve briefer, clearer, more general, and more effective summary statements about the problems of various disciplines than is possible with conventional algebraic notation, or with the *ad hoc* extensions that each discipline tends to develop to serve its special needs. At the same time, students can experiment with these APL expressions and execute them at a terminal without special instruction in data processing techniques, since they need no further translation.

This approach is so effective because of the underlying concept of 'a common language for man and machine'. For the sake of brevity, the language is symbolic for all frequently used functions (such as the well-known algebraic function +, -, ×, and so on), but the set of available symbols also includes non-mathematical functions.

Additionally, each user can create his own set of functions using descriptive natural language if he so prefers. To illustrate this with a trivial example, suppose a user prefers to write *PLUS* instead of the symbol + . He could then define a function such that *M PLUS N* yields the same result as *M+N* , thus making the symbol + completely transparent.

Indeed, where commonly accepted symbols for specific functions do not exist, we use exactly this method of naming functions with a plain-language description, as was shown earlier in the examples of the *CORRELATE* and *STANDARDIZE* functions. Thus, when one uses APL, the common language between man and machine becomes the language of man.

Let us now abstract some generalities from the example. The claim that APL can be used as an effective teaching tool is based on these premises:

* That key concepts of various disciplines may be represented as functions, and that these may become building blocks for more complex functions

* That it is possible to write computer programs so that they correspond directly to the functional concepts of a discipline

* That a language such as APL permits both a readable, formal definition of a function and a means of executing it, thereby allowing the accumulation of experience helpful for understanding it.

If functions can be written in such a way that they become directly executable on a computer, then the computer takes on a new role. To be able to encode a reasoned solution such that it can be interpreted by a computer means that the proposed solution has been effectively communicated to an independent (and impartial!) third party. As is well known, scientists have not always been good at this, although it is essential for communication between scientists and for providing repeatable solutions.

However, the greatest benefit for the learner comes from the process of encoding the solution in combination with the immediate feedback that is available from the computer: He can see at once whether his code is acceptable, and whether it behaves as he expects. This encourages him to explore and experiment with proposed functions to investigate their behavior. As a result, the learner can not help but gain deeper understanding and insight into his subject.

This also illustrates that the use of APL has benefits quite apart from its use as a computer language. There are significant advantages to using the notation, even if no use of terminals is made, because any language acceptable to a computer must necessarily be entirely unambiguous. As shown with the *CORRELATE* function, the notation makes the usual subscripting in algebraic expressions redundant. Also, all relevant variables and sub-functions are given self-explanatory names to aid the understanding of their meaning.

Finally, the required steps in the process are sequentially numbered. This not only enables the computer to interpret these statements, but also the human user. Generally, however, one would use a terminal to provide rapid and decisive testing of proposed solutions. This has the dual advantage of enforcing a much stricter discipline than would otherwise be available, and motivating students to write functions with the rigor required for computer interpretation.

HOW TO TEACH A CONCEPT WITH 'OPEN USE'

Typically, a student comes to understand a function from three kinds of experiences:

* Knowing how a function is defined (that is, a formal analysis of how its definition is composed of previously established more primitive concepts)

* Experiencing its properties. The intense, intuitive familiarity that accompanies its frequent use is generally built up from accumulated experience of the way the function behaves in a variety of contexts, and

* Seeing a particular function in its place as part of a larger system of functions.

Note, that conventional teaching relies primarily on the first kind of experience - formal analysis. Experiencing a function's properties is seldom carried beyond a few sample exercises because the amount of manual calculation becomes quickly prohibitive as the number of variables increases. For example, exercises on functions requiring inverses are seldom carried beyond a 3-variable example.

But when a computer carries out the drudgery of calculating, students can really "experience" the function because they can explore its entire range in a matter of minutes. Thus, an intercorrelation matrix from 20 variables can usually be obtained in the time it takes to display the matrix on the terminal, whereas it would require many hours of human calculating effort.

The availability of an APL-terminal, therefore, significantly changes the limit on both the quantity and the extent of the exercises which a teacher might want to give his students to let them experience the properties of a function. For example, on a computer one can obtain intercorrelations of *HEIGHT* and *WEIGHT* for a sample of 5 as easily as for a sample of 500 - provided the data are available.

In such a setting, one could approach an introduction to correlation by first assigning students an exercise to explore the *CORRELATE* function on their own. For instance, the class could be split into groups of, let us say, five students, and each group could be assigned to collect a series of their own personal data variables to be correlated - e.g., last year's grades, high school grades, family data such as number of children, own home versus rented home, their height, weight, waist, age, and so on. In addition, several randomly generated variables could be included.

The discussion of this exercise in class would naturally lead into all the problems associated with correlation, such as assumptions underlying the algorithm, its applicability to dichotomous and continuous data, spurious correlations among variables when using small N's, other algorithms used for calculating correlations (using different assumptions), and so on. The data from several groups could be combined to demonstrate the increasing stability of coefficients with larger N, and the disappearance of spurious effects among the random variables. This leads into significance testing.

Note that such an approach works well because it also relies on incremental learning. In other words, the only new thing to be learned is the function. The student needs to learn nothing about the data because it comes from his environment, and he is already intuitively familiar with the interconnections among most of the variables. Thus, he can check his correlation matrix to see if it makes "sense". He could not do this if both data and function were new to him (as is often the case with conventional exercises designed to facilitate marking).

INSIGHT AND THE CONCEPT OF FUNCTION

This example on teaching correlation illustrates that the ideas of a discipline are generally conveyed via two elements:

* the DATA: those attributes that we choose to note and measure, and

* the FUNCTIONS: the transformations of data from one representation to another, and the relationships between different sets of data.

The role of functions emerges from this as the second and third levels of behavior illustrated in Table 1. Functions and data interact with each other because the choice of the functions by which data is to be treated will influence the selection of facts worth observing, and vice versa. In many disciplines, particularly in the area of science, concepts can be expressed in the form of functions. The notion of function is surprisingly familiar: it is one of the few words with a popular usage quite consistent with its more formal definition. People use phrases like 'a function of time' in a way that they need not unlearn when they study mathematics.

In Iverson's notation, the meaning of 'function' embraces what we commonly call 'operators' in mathematics. These primitive

functions, as they are called in APL, can be easily represented by symbols (such as +, -, ×, etc.), and it is particularly convenient to do so when these symbols represent a unique or frequently used operation. Such functions are generally easily understood, and present no problems. More difficulties arise in understanding complicated functions, because our customary notation is not suited to express the components of such functions to show their structure in a 'building block' fashion. By overcoming this particularly troublesome problem, 'open use of APL' provides an excellent vehicle to assist students to gain insight into the topic under study, and thus enhance their understanding of it.

Taken together, a collection of functions imposes a structure on the understanding of a topic. For some time, Falkoff and Iverson have argued that a system of functions (or programs) can constitute a framework for a discipline (for instance, Iverson, 1966, page 53). Iverson has also suggested (1972, pages 230-242) that the language of science resembles informal speech in that each contains nouns (to refer to data, or observations) and verbs (to describe actions, or transformations). In APL, functions that describe how items of data are related and transformed might be thought of as active, transitive verbs. It is often possible to choose the name of a function so that it corresponds directly to terms used in less formal discussion of the same topic, and thereby to achieve a parallel between the structuring provided by the natural-language description of a topic, and its formal description as a system of data and functions.

This approach has been illustrated in the above example where the names of functions have been expressed in the form of verbs whenever a suitable choice was available (e.g., *STANDARDIZE* or *CORRELATE*). Input and output of these functions has been expressed as nouns. This convention tends to yield clearly understandable statements, as in

CORRELATE HEIGHT AND WEIGHT

Unfortunately, the vocabulary for expressing functions in natural language is not very complete, and as the complexity of the functions increases, one encounters situations requiring a compromise. For instance, we could not think of a suitable verb to name a function for calculating the variance-covariance matrix of a set of data. Yet, it is particularly in situations where natural language is inadequate as a vehicle for communication that the 'open use' approach provives important educational advantages, because APL has been designed as a language that permits clear and concise statements of procedure, to make them easier to think about as well as easier to evaluate by machine. The educational relevance of APL follows from its salient characteristics as a programming language. These are:

* A simple syntax consistently applied to all functions
* Simple rules for applying functions to arrays, so that arrays may be treated as wholes
* A rich set of primitive functions, including many from mathematics and logic, and new ones for manipulating arrays and non-mathematical data
* The possibility of extending the language by defining new functions in terms of those already defined (including functions the user has himself defined earlier).

In each field or discipline, the user may in effect extend the language to embrace functions appropriate to it. The user may build up definitions in a pyramid, using definitions from any level of the pyramid as terms in the definition of the next level (see Figure 1).

Since APL permits an unusually elegant treatment of arrays, a function whose definition applies to a single instance can often be applied (with perhaps little modification) to an array. This property greatly simplifies many expressions. APL in this regard resembles, and extends, matrix algebra. Operations on matrices (e.g., tables) are central to thinking about many topics, including large areas of business data processing. The 'open use of APL' approach exploits these features of APL for the benefit of the students.

To illustrate, consider the earlier assertion that the APL notation can easily be expanded from the univariate case to the multivariate case. Let us illustrate how this can be done, and how - in the process of so doing - the problem takes on a different connotation.

Consider the evaluation of the MEAN. An often demonstrated APL function for evaluating the mean is as follows:

```
      ∇ M←MEAN DATA
[1]   M←(+/DATA)÷(⍴DATA)
      ∇
```

This function definition uses the size-operator ⍴ which returns the size of its argument, in this case, *DATA*; further, the compression operator / which compresses its right argument to the next lower dimension using the function on its left. In the above case, +/ indicates additive compression (or summing) of *DATA*. Thus, statement [1] reads: *M* is assigned the sum of *DATA* divided by the size (number of elements) of *DATA*.

Note that this function definition works only when the structure of *DATA* is one-dimensional, that is, when *DATA* is a vector. In that case, ⍴*DATA* returns the number of elements *N*, and +/*DATA* returns the sum of the elements which is then divided by *N*. If *DATA* is a matrix or any other higher order array composed of multiple variables, then ⍴*DATA* returns the size of the array in the form of a vector containing one element for each dimension of the array. For example, the execution of

```
      ⍴ HEIGHT AND WEIGHT
5  2
```

indicates an array of 5 rows and 2 columns.

The compression operator (/) by default compresses its argument over the last dimension unless specifically instructed to do otherwise. In the case of a vector argument, the last dimension is also the first one, and thus it makes no difference. But in the case of a matrix, for example, it would compress the columns of the matrix. To illustrate with additive compression,

```
      +/ HEIGHT AND WEIGHT
204.3  180.1  187.5  170.5  184.1
```

compresses the matrix into a vector of row sums. This vector can obviously not be divided by the size vector as they are of unequal length. The above definition of the function *MEAN* is, therefore, not a good one, because it can only deal with one specific data-type (vectors), rather than being a generally valid function.

Let us now redefine the function *MEAN* as "the additive compression of the first dimension of an array divided by the size of the first dimension". Such a definition would treat not only data arranged in vector format (where the first dimension is simultaneously the last), but would apply to arrays of any dimension.

The following function is not limited by data-types:

```
      ∇ M←MEAN DATA
[1]   M←(+⌿DATA)÷(1↑⍴DATA)
      ∇
```

This definition is not substantially different from the earlier one. The modified compression operator ⌿ indicates that the default has been reversed: instead of compressing the last dimension (as with /), the ⌿ will compress over the first dimension of the argument. The ↑ is called the take-operator, and 1↑ indicates "take the first element of ⍴*DATA*". The additional notation required to write this generally applicable function is minimal. However, while the formulation of generally valid expressions in APL does not greatly increase the complexity of the algorithm, it does increase ones understanding of the problem. The redefined *MEAN* function corresponds exactly to its English language definition. This illustrates the close correspondence between precisely expressed natural language definitions, and their symbolic translation into APL statements. Natural language expressions can often be litterally translated into APL expressions.

The above function returns the mean of the first dimension of any array. In statistics, we generally arrange data in the form of columns, and then calculate the column mean, writing it at the bottom of the sheet below each column. In APL terminology, we compress the matrix over its first dimension, that is, rows are compressed into one single row. An example showing the execution of this function is given in the appendix, together with the multivariate definitions for all the functions represented by boxes in Figure 1.

Those readers who are experienced in using generally available statistical packages will certainly be impressed by the simple nature of expanding these functions from univariate to multivariate use. While the APL symbolism may at first appear strange, the meaning of these symbols becomes quickly familiar because most of them represent procedures which one commonly uses in pencil-and-paper work without explicitly realizing it. A case in point is the *AND* housekeeping function. It provides the possibility of concatenating data in tabular form similar to the way we do this on a sheet of paper where we simply add another column at the end of a data matrix when required. The APL notation provides a convenient and easily understood shorthand for such commonly used operations. The brevity of this shorthand is a powerful tool for describing problems and processes, but the symbols can be replaced when this is desirable for the sake of clarity. We believe that this point is well illustrated with the introduction of the housekeeping functions used in the statistics routines.

In teaching, APL functions should always be written in such a way as to make it as easy as possible for the student to understand the function, since he is expected not only to use the function but also to grasp the principle behind it. Each function, therefore, should not be a black box but a glass box, the inner mechanism of which is visible to anyone who cares to study it.

We call this 'open use' of the computer. In this context, the computer is a laboratory device - an arena for testing ideas. The task of the student is to develop functions for solving the problems of his discipline, and then to use these to gain experience in the behavior of the functions. He should be led by the teacher to find

solutions of general validity as was illustrated on the preceding page with the *MEAN* function. Since in statistics, the multivariate sclution includes the univariate, it is always a more general solution of the problem. Note that this solution could be achieved without the need to first define a univariate solution in order to lead the student from the simple to the complex, because the multivariate solution in APL is only marginally different from the univariate one.

To summarize, the attributes of 'open use of APL' that are valuable to education are:

* The importance given to proper definition of functions
* The formal consistency with which functions are treated
* The possibility of choosing functions so that they not only perform calculations, but also correspond pedagogically to the concepts of the discipline.

Since these concepts are expressed in the form of functions that are always 'open' to the student for inspection, modification, and experimentation, the approach was given the name 'open use of APL'.

IMPLEMENTING 'OPEN USE OF APL'

To implement 'open use of APL', access to one or more APL terminals is needed. The terminals may reside right in the classroom, where they can be used just like laboratory equipment, or, the school might set aside a special room for use by teachers and students. The terminals look either like typewriters, or like television screens with attached keyboards. They may have a desk-type computer built in, or they may be connected (sometimes via a telephone link) to a centrally located computer. The computer evaluates what a student types on the keyboard, then returns its reply either to the typing mechanism or the screen, as the case may be. It is necessary, of course, for the computer to have the APL language among its repertoire of programming languages.

Once an APL terminal is available, a teacher can use it immediately as part of his course as no courseware needs to be specifically prepared apart from normal lesson preparation work. The teacher's demonstrations of functions at work can be shown to large groups with the aid of television monitors. Together with his students, he can explore the most suitable ways of structuring and defining his discipline.

KNOWLEDGE REQUIRED OF THE TEACHER

Since the APL notation differs somewhat from conventional algebraic notation and introduces many new ideas useful in solving problems, teachers wanting to use 'open use of APL' need to unlearn some of the rules imposed by high school mathematics, and unders and the new concepts. Therefore, an introductory seminar to familiarize teachers with APL and the 'open use' approach is recommended.

REFERENCES

Berry, P.C., Bartoli, G., Dell'Aquilla, C., and Spadavecchia, V. APL and insight: The use of programs to represent concepts in teaching. IBM Technical Report (order number 320-3020), 1973.

IBM. An application guide to open use of APL in teaching science. IBM Technical Report (order number GE 156044), 1975.

Iverson, K.E. Elementary functions: An algorithmic treatment. Chicago: Science Research Associates, 1966.

Iverson, K.E. Algebra: An algorithmic treatment. Menlo Park: Addison Wesley, 1972.

Romano, A. The CSATA project: An instructional interactive model for science teaching. In Computers in the Instructional Process. Report from an International School, ed. Zinn, K.L., Refice, M., and Romano, A. Ann Arbor: Extend Publications, 1974.

Spence, R. Resistive circuit theory. New York: McGraw Hill, 1974.

APPENDIX

APL FUNCTIONS TO DEFINE STATISTICAL CONCEPTS

Assignment of variable names to data:

```
      HEIGHT← 156  138  145  129  141
      WEIGHT←48.3 42.1 42.5 41.5 43.1
```

Definition of arithmetic mean:

```
      ∇ M←MEAN DATA
[1]     N←ROWS DATA
[2]     M←(+⌿DATA)÷N
      ∇
```

If one wishes to expose students to matrix algebra at an early stage in the statistics course, one could use this elegant definition of the mean:

```
      ∇ M←AVERAGE DATA
[1]     N←ROWS DATA
[2]     M←(÷N)+.×DATA
      ∇
```

Definition of DEVIATION SCORES:

```
      ∇ DEVIATIONSCORES←DEVIATE DATA
[1]     DEVIATIONSCORES←DATA-(EXPAND MEAN DATA)
      ∇
```

Definition of VARIANCE:

```
      ∇ V←VARIANCE DATAMATRIX;D
[1]     D←DEVIATE DATAMATRIX
[2]     N←ROWS DATAMATRIX
[3]    ⍝V IS VARIANCE←COVARIANCE MATRIX
[4]     V←((⍉D)+.×D)÷(N-1)
      ∇
```

Definition of STANDARD DEVIATION:

```
      ∇ STD←STΔDEVIATION DATA;V;M
[1]     V←VARIANCE DATA
[2]     M←(⍴⍴V)⍴1
[3]    ⍝ STD IS 2. ROOT OF MAIN DIAGONAL OF V
[4]     STD←(M⌽V)*0.5
      ∇
```

Definition of STANDARD SCORES:

```
      ∇ ZSCORES←STANDARDIZE SCORES
[1]     ZSCORES←(DEVIATE SCORES)
            ÷(EXPAND STΔDEVIATION SCORES)
      ∇
```

Definition of CORRELATION:

```
      ∇ CORRELATION←CORRELATE DATAMATRIX;S
[1]     S←STANDARDIZE DATAMATRIX
[2]     N←ROWS DATAMATRIX
[3]     CORRELATION←((⍉S)+.×S)÷(N-1)
      ∇
```

The following are the three housekeeping functions used with this set of statistical functions:

```
      ∇ N←ROWS DATA
[1]     N←1↑(⍴DATA),1
      ∇


      ∇ MATRIX←EXPAND COLUMNS;RC
[1]     RC←(1,1≠⍴COLUMNS)/N,⍴COLUMNS
[2]     MATRIX←RC⍴COLUMNS
      ∇


      ∇ MATRIX←X AND Y;I;J
[1]     →((I←ROWS X)≠(J←ROWS Y))/6
[2]     MATRIX←(,⍉X),(⍉Y)
[3]     J←(⍴MATRIX)÷I
[4]     MATRIX←⍉(J,I)⍴MATRIX
[5]     →0
[6]     'LENGTH ERROR: ';I,J
[7]     MATRIX←⍬
      ∇
```

SAMPLE OUTPUT USING THE STATISTICAL FUNCTIONS

```
      HEIGHT
156 138 145 129 141
      WEIGHT
48.3 42.1 42.5 41.5 43.1

      DATA←HEIGHT AND WEIGHT
      DATA
  156    48.3
  138    42.1
  145    42.5
  129    41.5
  141    43.1

      MEAN DATA
141.8  43.5

      EXPAND MEAN DATA
  141.8    43.5
  141.8    43.5
  141.8    43.5
  141.8    43.5
  141.8    43.5

      STΔDEVIATION DATA
9.884  2.746

      CORRELATE DATA
  1        0.8861
  0.8861   1
```

IMPLEMENTATION OF DATA PROCESSING INSTRUCTIONAL TECHNIQUES

Jeff A. Schnepper
Rutgers College

Accounting is the art of classifying and recording financial information. In this century, the most important innovative tool developed for the accountant's workbag has been the introduction of the computer. With microsecond speed and input reliance accuracy, the computer has eliminated much of the pencil pushing drudgery previously necessary for processing accounting information. In the spring of 1975, I attempted to apply the computer to another function - the teaching of financial accounting. This paper is a report of that experiment, and the analysis of its results.

In the fall of 1974, I had taught two basic introductory courses with a total enrollment of 187. The second half of that course was taught in the spring with an enrollment of 163 for the two sections. Each of the two sections was of equal size. There was also no preregistration announcement as to any special activity to take place in either section.

When enrollment had been completed, I decided class one was to be taught in the traditional way but to divide class two. One half of that class, selected randomly, was also to be taught in the traditional manner. The other half of the class was to complete their weekly homeworks by utilizing our campus computer terminals which had been preprogrammed with the Pillsbury Package, a compilation of accounting programs.

In both terms, fall and spring, I had been the only instructor of introductory accounting. I therefore was saved the additional variables of new students, or different introductory presentations. Both classes were taught in exactly the same way and, as the computer students were identified only by student number, no in class differentiation was possible.

The Total Class comparisons can be seen in Table I. Although Class One began with clearly the "better" students (44% of Class One had been A-B students compared to 29% of Class Two), the students of Class Two had overtaken them by term end (81% Class Two A's and B's compared now to a relatively constant 44% for Class One).

Two possible explanations could account for the amazing growth of Class Two. The first is the "Hawthorne" effect. In 1924, the General Electric Company conducted an experiment at its Hawthorne plant to improve production. The women in the experiment did the same work they had done before, but experimenters changed the lighting and the pace. The changed the number of breaks, the snacks, etc. Yet no matter what they did, production improved. It was the impression of the workers which had made the most difference. They were aware that they had become part of an experiment and therefore had performed better.

While I am confident that the Hawthorne effect did have some input on the success of Class Two, it would not account for the relative stability of Class One's grades. Perhaps the students of Class One feit themselves too far removed from the experiment to be significantly effected.

The second possible explanation for the increased grades of Class Two could be the effect of computed augmented instruction. To investigate this, we must analyze the results of Table II.

The non computer section of Class Two increased its number of A-B students from 13 to 28, 32% to 68%, a 36% increase. The computer section though, increased from 11 to 38, 27% to 93%, an increase of 66%!! Plainly, while the Hawthorne effect might have accounted for a portion of this growth, it would not alone explain this significant a difference.

What further conclusions can be drawn from the above results? While I would like to say that the numbers prove that computer augmented accounting instruction promoted a better understanding of the subject, one must recognize that this was a single experiment with a relatively limited sample size.

What I can say though, is that the results clearly indicated, in this case, an increase in interest and understanding of not only accounting, but the use of the computer as well. In an era where time and cost are so closely related, where the future of corporate accounting will surely be in the area of computer utilization, this alone should make computer augmented accounting courses a valuable and important pedagogical tool.

Table I

Total - Class I

A	B	C	D	Total	
19	17	42	3	81	First Term
21	15	45	0	81	Second Term
2	<2>	3	<3>		Difference

Total - Class II

A	B	C	D	Total	
5	19	51	7	82	First Term
23	43	16	0	82	Second Term
18	24	<35>	<7>		Difference

Table II

Class II - Computer v. Non Computer

Computer

A	B	C	D	Total	
1	10	24	6	41	First Term
15	23	3	0	41	Second Term
14	13	<21>	<6>		Difference

Non Computer

A	B	C	D	Total	
4	9	27	1	41	First Term
8	20	13	0	41	Second Term
4	11	<14>	<1>		Difference

GRAPHICS IN THE GEOGRAPHY CURRICULUM

FRANK T. ALDRICH

Geography Department, Arizona State University, Tempe, Arizona 85281

ABSTRACT: Geographers have a great preoccupation with maps. Since before the time of Plato, they have made, used, written about, and discussed maps, spatial distributions and spatial relationships. Then, as today, a map represents a theoretical construct that forms the basis of geographic thinking and research. The uses range from data gathering and analysis to synthesis and information display. Computers have, of course, introduced many refinements and improvements into geography curricula and research: noteably in Computer Assisted Instruction and computer mapping and graphics. As in many diciplines, the C.A.I. has met with only limited success. However, in the area of computer mapping this tool has allowed great strides to be made and offers the potential for adding a possible new dimension to spatial research and analysis techniques. This presentation reviews the benefit of computer graphics in the field of geography and the future potentials of using high technology analysis and display capebility. The classroom experience at Arizona State University is used as a basis for the conclusions that have been drawn.

ECONOMIC THEORY AND THE COMPUTER
THE MACROECONOMIC SIMULATION

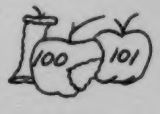
Kathleen A. Powers

American Graduate School
of International Management
Glendale, AZ. 85306

ABSTRACT: The economics professor is especially vulnerable to student criticism that his carefully constructed world bears little or no resemblance to reality. Course requirements, student backgrounds, and semester time constraints force him to dwell upon theoretical minutiae to the virtual exclusion of real applications. Thus the well-programmed computer simulation becomes a welcome aid to the instructional process. The professor may devote his lecture time to the important theoretical concepts while the students are exposed to proper application techniques at the computer. This paper is about one such union of theory and practice: the AGSIM macroeconomics course and its MACRO simulation.

As a science economics is often misunderstood. Economics is supposed to explain how the economy works. Its theories are often expressed with the same certainty and statistical precision of chemistry and physics; but its subject matter--human behavior, individual and/or aggregative--is inherently unpredictable and changeable. People, whether in labor, business, politics or whatever, expect economists to be able to analyse an economy's strengths and weaknesses, to diagnose any difficulties, and to propose remedies for problems. Then, during such periods as the early/mid-seventies, the public discovers that economists cannot agree on the problem much less advance any "sensible solutions." Rather the "specialists" either hide behind what seems to be theoretical conjecture or make suggestions which do not shed any new light on the current situation.

Not an impressive track record! For the classroom teacher the situation worsens. Much time must be spent teaching the various theories of economics. A highly specialized vocabulary is established. First, the economy is broken into small parts, and the students learn it is important to distinguish between the utility of eating your 100th apple and the utility of eating your 101st apple. Then, the economy is taken as a whole, and the students learn that individual decisions, whether made by people or corporations, do not affect the economic outcome. Naturally, the students "discover" that economists are involved in a number of ivory-tower games, blind to the chaos and reality around them, and content to let the real world handle its own messy problems since they rarely fall within the very strict confines of "economic theory."



It is at this point that the computer enters the economics classroom and helps the students discover that all the dusty, minute detail of theory, despite some obvious shortcomings in certain areas, does reflect REALITY--does help a person understand what is going on around him and why. The computer simulation allows the professor freedom to explore with his students both the theory and practice of economics. This factor is especially important at American Graduate School of International Management, Glendale, Arizona.

The AGSIM student body has varied backgrounds both educationally and experientially, but less than one third of the students has had much exposure to economics or business courses of any kind. Nevertheless, the school produces well-rounded, qualified candidates for management positions in international (and domestic) business and does it in three semesters. The challenge is fantastic! AGSIM's Professor must provide insight and exposure to the intricacies of economic theories, AND he must enable the student to grasp the reasonable use of those theories when he is out in the real world. When it comes to helping students learn the differences between Monetarist and Keynesian economics and understand how to use economics as a tool in their future decision-making, whether as businessmen, politicians, bureaucrats, or voters, the MACRO simulation becomes invaluable.

The AGSIM MACRO simulation has been especially adapted and tailored to our macroeconomics course[1] content and needs from the more general simulation developed at the University of Wisconsin.[2] The basic economic model used is the Keynesian IS-LM model for a two-country world with fixed (1=1) exchange rates. The program is versatile in that we can also approximate a Monetarist model by changing several parameters. The software was developed by our computer department. The hardware is the Hewlett - Packard 9830A.

MACRO is programmed in such a manner that the students input the absolute values of nine economic

[1] WB 423--Monetary and Fiscal Policies

[2] Peter H. Lindert, MACRO: A Game of Growth and Policy, (New York: Holt, Rinehart and Winston, Inc., 1970).

variables as inputs (Exhibit 1) in the simulation. The computer output (Exhibit 2) presents the national accounts for the two trading countries, ininclueing the balance of payments and the interest rate. tape records of each period's activity provide histories of the countries' development.

As a visiting professor I had the privilege of teaching WB 425 to some 181 students over a period of five semesters.At the beginning of each semester I told my students to face their study of macroeconomic theory in the same manner that they faced language studies -- to concentrate on learning the vocabulary of macroeconomics and the "syntax and grammar"evident in the basic principles and behavior patterns we would study. I also encou aged them to discover and remember the skeleton, or framework,of macroeconomic theory and model building which they could, then, duplicate in any socio/economic situation.Our class emphasis would be on the US economy. We would also find out how the accuracy and scope of behavioral assumptions are crucial to the development of economic theory and its practical implications--for planners in business as well as in government. However, the computer work would give them the opportunity to sharpen their understanding of theory and to test their abilities as economic planners and decision-makers. These points were reiterated at appropriate lecture/discussion points throughout the semester.

The first four semesters I taught WB 425, the theoretical model was developed in class before the students began their computer simulations. My idea was to prepare them for their policy-making roles. The fifth semester we began the simulations after only the barest of introductions to macroeconomic theory. Despite their initial shock, students in this latter group took no longer to become familiar with the computer model than those in the former group. Moreover the"fifth semester"students seemed to understand better and to anticipate the theoretical development as the course proceeded. Therefore, I believe the longrun maximum benefit to the majority of students is achieved by starting the computer work without delay.

LOGISTICS for the computer work were straightforward. Each class was divided into country teams of three or four students.Then the "countries" were paired to form the worlds called for in the computer program. These teams participated in three or more simulations during the semesert. The first simulation assignments were as follows:

1. Determine and turn in written goals for your economy.
2. Turn in input cards for one period at a time. Get and evaluate computer printouts. Make new inputs. A minimum of two computer runs per week is required.
3. Every two weeks or so (four to six periods) turn in a written progress report detailing your course of action, its purpose and success,and any changes in country goals.

The second simulation was carried on simultaneously with the first, with the same worlds. It merely gave the students a chance to compare strategies under (or against) different foal situations and/or different socio/political philosophies.

In order to reduce the"end of world" technique or "last minute coup" philosophy I did not tell the classes how long any of the simulations would run. Generally the first two simulations ran between ten and sixteen periods each. The rest of the semester was spent on one or more of the following variations:

Recall the Original version--simulations 1 and 2
Each team decides one input set for each simulation. (Different goals are encouraged.)
Country teams A, B, C, and D comprise Worlds AB, AB, CD, CD.

American Graduate School Macro Simulation Computer Model

TEAM B (AGAINST TEAM A) GAME 1 PERIOD 2 DATE 3/1/7c

10.2 1.
.24 2.
.26 3.
11.7 4.
34 5.
.19 6.
1 7.
19 8.
7 9.

American Graduate School Macro Simulation Computer Model

TEAM A (AGAINST TEAM B) GAME 1 PERIOD 2 DATE 3/1/7c

10.5 1. *Employment (may not exceed labor force)
.21 2. @Personal tax rate (a percent)
.25 3. *Real fixed investment (purchases of capital)
11.3 4. Labor force size (+.3> change >-.2)
35 5. @Money supply
.17 6. @Profits tax rate (a percent)
1.01 7. *Price level
18 8. @Real government spending (>8)
7.2 9. Wage rate ($ per employment unit)

@ Indic

Remember changing M will change interest rates; nominal production & cost determine profits; real production & real sales affect changes in inventory.

@ Indicates government decision; *Business decision; Blank, labor decision

Exhibit 1. Sample Student Input Cards

INTERNATIONAL ECONOMIC MACRO SIMULATION

COUNTRY A1 TIME PERIOD 2

LABOR ACCOUNT

W	N	W*N	i*B	PI	T PI	DI
7.2000	10.5000	75.6000	15.1251	90.7251	19.0523	71.6728
C/P	C	Sh	Acc Sh	M + Mf	B	Mf
66.7706	67.4383	4.2346	237.3546	34.4250	202.9295	11.3000

GOVERNMENT ACCOUNT

G/P	G	i*Bg	t PI	T PI	t profits	T profits
18.0000	18.1800	7.8067	0.2100	19.0523	0.1700	2.2036
T PI+Tprofits	Bud.def	Cum def	M	Bg	STC+LTC	BofP
21.2558	4.7309	139.3768	35.0000	104.3768	0.5482	-0.9491
Acc BofP = Mf						
-0.5750						

BUSINESS ACCOUNT

i*Bb	dep*Kt-1	W	N	W*N	W*N+i*Bb+dep*Kt-1	profits
7.3184	10.7000	7.2000	10.5000	75.6000	93.6184	12.9622
Tprofits	profits-Tprofits	P	y	Y	If/P	G/P
2.2036	10.7586	1.0100	105.5253	106.5805	25.0000	18.0000
C/P	(X-M)/P	Sales/P	chg Inv/P	K	Inv/P	Inv/P+K
66.7706	-1.4974	108.2732	-2.7479	228.3000	12.4252	240.7252
Assets	Net Worth	Bb				
240.8400	141.8318	99.0089				

INTEREST RATE 0.070116908

24

207

INTERNATIONAL ECONOMIC MACRO SIMULATION

COUNTRY B ℓ TIME PERIOD 2

LABOR ACCOUNT

W	N	W*N	i*B	PI	T PI	DI
7.0000	10.2000	71.4000	12.6373	84.0373	20.1689	63.8683
C/P	C	Sh	Acc Sh	M + Mf	S	Nf
61.8947	61.8947	2.7737	234.9837	34.5750	200.4087	11.4000

GOVERNMENT ACCOUNT

G/P	G	i*Bg	t PI	T PI	t profits	T profits
19.0000	19.0000	6.8216	0.2400	20.1689	0.1900	2.7893
T PI+Tprofits	Bud.def	Cum def	M	Bg	STC+LTC	BofP
22.9583	2.0634	130.6886	34.0000	96.6886	-0.5482	0.9491
Acc BofP = Mf						
0.5750						

BUSINESS ACCOUNT

i*Bb	dep*Kt-1	W	N	W*N	W*N+i*Bb+dep*Kt-1	profits
6.6156	10.8000	7.0000	10.2000	71.4000	88.8156	14.6807
Tprofits	profits-Tprofits	P	Y	Y	If/P	G/P
2.7893	11.8913	1.0000	103.4963	103.4963	26.8000	19.0000
C/P	(X-M)/P	Sales/P	chg.Inv/P	K	Inv/P	Inv/P+K
61.8947	1.4974	107.5928	-4.0957	231.2000	15.9624	247.1624
Assets	Net Worth	Bb				
247.1624	143.9985	103.1719				

INTEREST RATE 0.063390658

Figure 2. Sample Two-page Computer Print-out For Input Cards in Figure 1.

Variation I--simulations 3 and 4
Similar goal and input conditions as Original. Teams A,B,C,and D form Worlds AC,AD,BC,and BD.

Variation II--simulation 3 or 5 (This is an attempt to approximate conditions and problems that arise in the multinational--in a three-country sense--world trade situation.[3]
Each team selects goals and makes one input set which the computer runs against each of two other teams' input sets.
Alternative 1: Teams A, B, C, and D comprise Worlds AB, BC, AD, and CD.
Alternative 2: Teams A,B, and C form Worlds AB, BC, and AC.

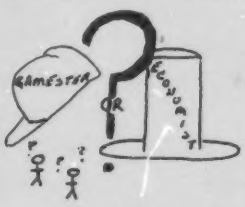
Although we developed the basic macroeconomic model, algebraically and graphically, during the first five or six weeks' lectures,the students were never told any of the parameter values used by the computer. Therefore,in order to make more intelligent plans and decisions,a number of students tried estimating the parameters by using various scatter diagram,regression analysis, or input/output study methods.It was exciting to see students(who claimed "Economics is out in left field!" at the beginning of the semester)employing various economic research techniques to help themselves become more efficient policy-makers.

The crucial importance of accurate behavioral assumptions in the policy formulation process was also clearly portrayed in the simulations. Sometime between periods eight and twelve,I would change the parameter set.[4] Such parameter changes were usually made after the class received a "News Leak," a leak might be about changing technology, war or emphasis on quality or it might just be about some expected change in economic behavior. Then, within a week, each world operated under its own, unique economic system.

The students were graded on their abilities to develop sound economic goals and to manage their economies toward that goal achievement. The parameter changes and simulation variations were instituted to keep the computer exercise one in economic logic rather than one of simple gamesmanship.

STUDENT RESPONSE was enthusiastic. Sometimes individual team members took total control of each sector of their economy; other times the team operated as a triumvrete over the whole economy. The latter were generally either pro-business or pro-labor, but one team was totally pro-dictators. (It should be noted that although the students looked forward to beginning the simulations, they were extremely hesitant to establish goals and to make their first input decisions. They approached the simulation as if it were a game like tennis,soccer, or backgammon. Some begged for simple, clear-cut rules, standards, criteria, or anything; all wanted more specific guidelines. I provided none.) Although some of their initial goals were too vague, and others, trivial, they had begun to correct this situation themselves, by the time their first progress reports were due. My primary role was that of consultant rather than professorial authority. I pointed out obvious errors in economic logic--one cannot be "fighting inflation" if production is increasing and prices are constant--and in model manipulation--changing the labor force size does not automatically change consumption. (My second role was that of chief computer operator.) The students were free to exercise their native intelligence and to develop their economic awareness to the fullest.

Student response to program changes generally indicated the simulation was becoming a game to them rather than an exercise in practicing economics. Thus, the changes were necessary and valuable in maintaining the latter as well as in revealing problems economic analysts face in "real world" situations and in increasing student understanding of the behavioral limitations inherent in any economic theory.



The students also developed some insight into the reasons for the sollusion among various economic groups, for the establishment of various trade barriers (None are allowed in MACRO.), and for the restrictions the very existence of trade puts on domestic policies. For example, Teams W and Z were one world in simulations 1 and 2. Team W chose to be a free enterprise economy in each simulation. Team Z, however, decided to operate at extremes. In the first simulation Z established a socialistic system; in the second , a laissez faire system. Due to the different trading patterns which developed, Team W found itself forced to make very different input decisions as the simulations progressed, and Team Z found it very difficult to determine "the two philosophies" governing Team W's activity

Therefore,the students realized the importance of economic awareness, and goals, to all decision-makers in the real world.More importantly, perhaps, they learned that the ability of economists to analyze and to provide solutions for world problems is significantly affected by the ability and the willingness of world peoples to behave in a similarly rational manner regarding the important factors of economic life.These facts became even more apparent in the "three-country" simulations of Variation II.

In summary,although it is still far from reality in every detail, the MACRO simulation is an invaluable tool to enable students of economic theory to put it to practice in their own lives. It promotes a greater appreciation among the students for the usefulness of that theory in decision-making.It provides a better understanding that variances between economic theory and real world practice are often the result of the inherent qualities in mankind rather than of the theoreticians' ineptitude.

---

[3]Due to programming constraints, this is the closest approximation we have to a three-country world. It is artificial in that real world economies do not have two different balances of trade or inventory levels, etc. during any one period.

[4]Parameters may be changed either singly or in toto. Different worlds may have different parameter sets without affecting computer time.

A STUDY IN THE SELECTION OF PREDICTORS FOR SUCCESS IN ELECTRONIC DATA PROCESSING COURSES

Dr. James A. Robb
School of Technical Careers
Southern Illinois University
Carbondale, Illinois

ABSTRACT: The problem was to determine personality factors which could be used for the counseling of students interested in a career in the programming of electronic computers. The first hypothesis was that the Sixteen Personality Factor Questionnaire (16PF) scores considered singularly and cumulatively in various selected combinations had no relationship with a student's being more successful academically (A and B course grades) as opposed to less successful (other grades and withdrawals) in an Electronic Data Processing (EDP) course. The second hypothesis was identical to the first but used the American College Testing (ACT) scores. The 1973 study involved 76 EDP students with 16PF scores and 60 students with ACT scores available who were enrolled in four EDP courses during the 1973-74 school year.

Each predictor variable was tested singularly and in combination with all other variables within its group using the stepwise regression analysis of variance method of the SPSS program "Regression." None of the 16PF variables taken singularly had a significantly high correlation to be considered significant at the .05 level. However, the group of the four variables Q_4, M, I and Q_1 had an F-ratio of 4.244, well above the 2.51 value needed to be significant at the .05 level.

All of the ACT scores except Natural Sciences were found to be significant predictors of the criterion, with mathematics having the highest correlation, .3950. These findings led to the conclusions: (1) None of the 16PF scores could be used singularly as a predictor of success in EDP, but when the four variables Q_4 (Relaxed vs. Tense), M (Practical vs. Imaginative), I (Tought-minded vs. Tenderminded), and Q_1 (Conservative vs Experimenting) were considered collectively as a group, they were considered significant predictors. (2) Three of the ACT scores, Mathematics, Social Studies and English could each be used as predictors of success in EDP. Also, the ACT scores may be used as a group in predicting success in EDP.

The 1975 follow-up study involved 140 students in several different EDP courses and used only the 16PF scores for these students. The ACT scores were not included in the follow-up study. The statistical procedure for the 16PF scores in the second study was the same as for the first study plus an additional computation of the correlations using four levels of success (A, B, C, & Other) instead of only two (Successful vs. Unsuccessful).

The findings of the second study included: (1) The correlations of each of the 16PF scores with EDP success were quite different from the correlations found in the first study except for Q_4, which did have a significant correlation with the two level (S/U) indication of success. However, (2) when Q_4 was correlated with four levels of success (A, B, C & Other) it was not significant. See Table 1 (Revised). The 16PF score regression analysis for the follow-up study was also computed in two ways as shown in Tables 2a and 2b. (3) When student success was indicated in two levels (S/U), Q_4 was significant with N (Forthright vs. Astute) as the next best but not significant predictor. However, (4) when success was indicated in four levels (A, B, C & Other) B (Intelligence) was found to be significant with Q_3 (Undisciplined vs. Controlled) as the next best predictor.

We may conclude from these findings that students with above average mathematical skills (ACT MA) or who are more tense (Q_4) will probably make better computer programmers when only two levels of success are indicated. However, when four levels of success are indicated, intelligence (B) is the best predictor of relative success.

CHAPTER I - INTRODUCTION

Background Information

Today, management information systems and modern computers with operating systems have demanded greater knowledge and skill from the computer programmer. As both the needs of industry for computers and the computers themselves have become more complex, the need for the highly capable programmers has increased.

It is very important that a student who is considering a career as a computer programmer be counseled carefully with regard to the duties and responsibilities of a programmer as well as with regard to his own potential. Further, and the premise of this study, a student's total personality must be taken into con-

sideration, not just his general intelligence. Personality traits which may influence his adjustment to the responsibilities and activities of a position as a programmer should be examined.

Over a period of about ten years, approximately sixty percent of the students who enrolled in the two year training program in electronic data processing at The School of Technical Careers at Southern Illinois University did not complete the program. Many of the community colleges in Illinois offered an associate degree in electronic data processing and they too experienced a relatively high dropout rate.

Many students recognized when they were not obtaining a satisfactory level of attainment and withdrew from the program voluntarily by the end of the first term. Others continued persistently in the program with a minimumly acceptable grade or until they were suspended by the University for low grades. Some students seemed to accept their predicament maturely and proceeded to select an alternate course of study while others underwent a very traumatic experience of failure.

There is a definite need for improvement in the counseling of students entering or intending to enter an electronic data processing career training program. Generally, when a student wanted to know his probability of success in EDP, there were two basic alternatives open to him. Either he could have enrolled in an introductory course and found out from experience how well he could learn and apply the fundamentals of EDP or he could arrange to take one or more of several aptitude tests that were available to him through counselors.

The first alternative, taking the course and "giving it a try," allowed the student with average abilities to find out for himself what was generally involved in computer programming; however, this method was both expensive and time-consuming. Further, it exposed the student who did not do well to a feeling of failure and frustration which could have been avoided by better counseling.

The second alternative, taking an aptitude test, might have indicated whether or not the student had the general ability to learn programming, but would not have indicated how well he might have liked it. Many students with high academic ability learned programming readily but, on the other hand, some of them tired of it soon after they learned it. As soon as the challenge and newness were gone, they often developed an aversion to the tedious precision of the task and no longer enjoyed their work. This type of aptitude test failed to identify the student who did only moderately well on the aptitude test but who eventually became an excellent programmer because of other personality factors and who usually did not get bored with the daily routine of a programmer.

Problem Statement and Hypotheses

A statement of the problem was to determine personality factors which could be used for the counseling of students interested in a career in the programming of electronic computers. Since aptitude and achievement tests alone did not identify personality characteristics which might also have been considered in predicting success or failure in EDP training, a different instrument was needed to identify them.

The primary purpose of this study was to identify specific personality traits as indicated by the Sixteen Personality Factor Questionnaire (16PF) which might have been possessed by successful versus unsuccessful students of computer programming. See Appendix B for a brief description of each of the 16PF factors. The study was expected to provide data which could have been useful in career counseling.

The first hypothesis for this study was that the 16PF scores considered singularly and cumulatively in various selected combinations had no relationship with a student's being more successful academically (A and B course grades) as opposed to less successful (other grades and withdrawals) in an EDP course. This hypothesis was tested at the .05 level of significance.

The second hypothesis for this study was that the ACT scores considered singularly and cumulatively in various selected combinations had no relationship with a student's being more successful academically (A & B course grades) as opposed to less successful (other grades and withdrawals) in an electronic data processing course. This hypothesis was tested at the .05 level of significance. Note: When the second study results were reviewed and found to be somewhat similar to the first study, it was decided to make another regression analysis with the criterion variable (grades) divided into four values (3, 2, 1, 0) instead of only two (1, 0).

Limitations of Study

The study was limited in several ways:

1. The time period was limited to the 1973 fall term (N=76) and the 1975 fall term (N=140).
2. The students were those enrolled in several EDP classes at the School of Technical Careers of Southern Illinois University at Carbondale.
3. The data involved were ACT scores, the 16PF scores, the EDP course letter grades, and enrollment records of the students.
4. In both studies course letter grades of A and B, as determined by each course instructor, were used to identify successful (value=1) students and all other grades or withdrawals identified the unsuccessful (value=0) students.
5. The follow-up study also included computations with values A=3, B=2, C=1 and all others=0.

Literature Review Summary

A literature review was completed to identify other studies supporting or related to this study, not only for the purpose of examining the results of the studies, but also to review the instruments and the procedures used in the studies. For brevity in this report, the findings of this review have been omitted.

In most of the studies examined in the review, some part of the 16PF inventory was found to be a useful predictor for the activity indentified in the study. Several of the studies used the ACT scores and two studies involved students in computer programming courses, but no study was found that included the 16PF test, ACT scores and an EDP course. In general, the best predictor of future success was found to be

past achievement.

The 16PF test was widely used as an instrument for identifying personality traits as attested to by these studies and by the more than 650 references to it in the Seventh Mental Measurements Yearbook; and, the ACT scores were extensively used as a measure of achievement. Thus, this review of related literature did support the selection of the 16PF test and the ACT test as appropriate instruments for this study.

CHAPTER II - PROCEDURES FOR THE STUDY

Selection of the Instruments

This study was designed to examine two readily available instruments, one for cognitive and one for noncognitive personality traits, in order to identify personality factors which might be used in the counseling of students to predict success in EDP training.

The general premise was that both basic intelligence and selected personality traits are involved in the successful performance of the functions or a programmer of business applications on modern electronic digital computers. Further, it was accepted that the two common terms "aptitude" and "interest" are essentially included in the terms intelligence and personality traits respectively. These two categories may be also referred to as cognitive and noncognitive characterisitics and the need for two separate instruments was anticipated.

The selection of the instrument that would indicate a student's aptitude, which was largely cognitive and based on past achievement, was not difficult. The one instrument that was widely used and the results of which were readily available to counselors was the American College Testing (ACT) examination. It was normally taken by high school students considering attending college, and the scores were sent to the colleges selected by the students. Thus the selection of the ACT scores as one instrument for this study was easily completed.

The selection of the second instrument, the one to measure personality traits or interests, was not so easily accomplished since no psychological or personality tests were commonly given to college students nor made a part of their admissions records. The selection of the psychological instrument was based upon these criteria:

1. Readily available to and easily administered by counselors without special training,
2. Could be administered in one hour or less on an individual or group basis without requiring a special setting,
3. Could be manually scored in a few minutes, and
4. The instrument must have been generally recognized and accepted among educators.

An instrument meeting these four criteria could be generally used by high school or college counselors.

Several dozen psychological and interest inventories were considered mostly by reading the description and the reivew presented in Tests in Print and in the Seventh Mental Measurements Yearbook. Many of the tests examined such as the Rorschach or the Thematic Aperception Test required that the counselor have special training in the administration and use of the test. The Sixteen Personality Factor Questionnaire (16PF) was selected as the most appropriate personality inventory for this purpose when all of the above criteria were considered. It was chosen to be used in this study as the instrument for the prediction of success in EDP training based on personality traits.

Brief descriptions of each of the 16PF traits are in the test form in Appendix B and a detailed description of them may be found in the 16PF Handbook and the Manual for the 16PF Inventory. The 1972 edition of the Manual also assisted in the final selection and use of the 16PF instrument, and the review of related literature supported the selection.

Selection of Subjects

At the time of the study students could enter the introductory EDP course from any of several schools or majors. Normally only the EDP majors took EDP courses beyond the introductory course, but advanced classes were open to any student with the prerequisites.

In order to obtain the largest possible number of subjects for the study within the School of Technical Careers and still maintain good control of the study, the following students were selected for the first study:

1. Students in the first term of the associate degree curriculum for business programmers. This group normally had a high dropout/failure rate.
2. Business Education students in either their first or second term of EDP course work. These students were required to take one course and additional courses were optional.
3. Students taking advanced EDP courses. All of these students were EDP majors and the normal dropout rate from these advanced classes was very low.

This mixed group served as an advantage in these studies since it provided an opportunity to examine the personality traits of the advanced students as well as those of the beginning students. For the follow-up study each of the EDP programming classes taught by the School of Technical Careers was contacted and students were encouraged to participate. A total of 140 students from 7 different sections of 3 different courses participated in the study. Most of these students were juniors or seniors who were taking their first EDP course. The course content was FORTRAN for business applications.

Implementation of the Study

In 1973, two major types of data were collected for the students involved. One part of the data, the ACT scores, was historical and was available for sixty of the subjects within the student record system of the University. The remaining seventeen students had no ACT score on file with the University indicating that they had not taken the test.

The other data, the personality factors, were not available within the University record system and therefore were obtained directly from the students. A specific class period was scheduled for each of the groups previously identified for the 16PF Questionnaire to be administered. Any student not attending class that period was asked to complete the questionnaire at his earliest convenience. Seventy-six students completed the questionnaire.

A brief explanation of the purpose of the study preceded the completion of the forms. It was important that good student rapport be established to encourage each student to prepare his form as truthfully as possible. The administration of the test, because of the nature of the instrument, was informal but uniform among the different groups.

After all of the available raw data were collected and the questionnaires evaluated, they were checked and recorded in a format appropriate for the statistical procedure.

The analysis of each student's personality profile was shared with him. Almost unanimously, the students who saw their 16PF scores indicated that the profiles were in agreement with their own self-images. Thus the students' reactions to the profiles were very positive.

In the 1975 study the ACT scores were not used because the primary area of interest was the personality profile. The 16PF questionnaire was administered as before except that mark sense forms were used for optical scanning. The 16PF scores were punched into cards, student grade values were coded in two ways so that two analyses could be made. In the first method, course grades A and B were weighted 1 and all others were weighted 0, as in the 1973 study. The second method weighted A=3, B=2, C=1 and all others=0. Thus the results of the 1975 study could be compared with the results of the 1973 study as well as permitting a comparison of a dichotomous criterion variable with one containing four values or weights.

Statistical Procedure Selection

Southern Illinois University at Carbondale had a large digital computer available for research studies. Several statistical programs were a part of the program library for the computer system. The subprogram "Regression" of the Statistical Package for the Social Sciences (SPSS) and the computer were used in the statistical analysis of this study.

The "Regression" program, using a stepwise regression analysis procedure, provided a comprehensive treatment of the data. This program selected and computed statistics for the most significant variable of the group being processed, then identified and selected the next variable which, when used with the previously selected variable(s) provided the best combination of variables as a collective predictor. The 1970 SPSS reference manual on page 180 states that:

> Stepwise regression is a quick and efficient method which provides a near-optimum solution to the problem. The method recursively constructs a prediction equation one independent variable at a time. The first step is to choose the single variable which is the best predictor. The second independent variable to be added to the regression equation is that which provides the best prediction in conjunction with the first variable. You then proceed in this recursive fashion adding variables step-by-step until you have the desired number of independent variables or until no other variable will make a significant contribution to the prediction equation.

The following statistics were calculated by the program:

1. The mean and standard deviation of each of the predictor variables for both groups of students in the 1973 study. (See Appedices E and F.)
2. The correlation coefficient of each of the predictor variables with the criterion variable for each group of students. (See Tables 1 and 3.)
3. The correlation, variance, F-ratio, and degrees of freedom for each of the progressive models selected in each group of students. (See Tables 2, 2a, 2b and 4.)

The two full models used for the multiple regression analysis, one for each of the student groups, were:

$$Y_{f1}=a_0U + a_1X_1 + a_2X_2 + \ldots a_4X_4 + E_1 \text{ for ACT scores}$$

$$Y_{f2}=a_0U + a_1X_1 + a_6X_6 + \ldots a_{21}X_{21} + E_1 \text{ for 16PF scores}$$

Y was the criterion variable.
a_0, a_1, etc. were the regression weights.
X_1 - X_5 were ACT scores for each student. (X_5, the composite score was not used.)
X_6 - X_{21} were the 16PF scores for each student.
E_1 was the error term.
U was the unit vector.

Each predictor variable was tested singularly and in combination with all other variables using the stepwise regression analysis of variance method of the SPSS program described earlier in this chapter.

CHAPTER III - STATISTICAL ANALYSIS OF DATA

Outline of the Statistical Procedure

The data collection procedure previously described provided the input data for analysis. Student data were divided into two sets for purposes of processing and analysis:

1. Students with 16PF scores, N=76, with 50 being successful and 26 being unsuccessful in their respective EDP courses.

Note-1a. In the follow-up study, N=140 with 93 being successful and 47 being

unsuccessful. Note that the ratio of successful/unsuccessful is the same.

1b. A second run of the follow-up study used four levels of success with N=140, A=43, B=50, C=29 and all others=18.

2. Students with ACT scores, N=60, with 39 being successful and 21 being unsuccessful.

A tabulation of all of the input data for the first study comparing the successful versus the unsuccessful counts for each predictor variable was made and the results of these tabulations are shown in Appendices C and D. This tabulation was not made for the 1975 study.

The SPSS program "Regression" was used to compute the mean and standard deviation of each of the variables for both groups. Then the difference between the means was manually calculated and entered into the tabulation in order to more easily compare the successful students with the unsuccessful students. These data are shown in Appendices E, F, G and H.

The results of the correlation and regression calculations are shown in the following tables. Tables 1 and 2 contain the correlations and regression data for the 16PF test scores while Tables 3 and 4 contain the correlations and regression data for the ACT scores.

Correlations of the 16PF Test Variables

The correlations of the 16PF scores with the criterion variables of success in an EDP training course are shown in Table 1 (Revised). None of the 1973 variables exceed .226, the value required for significance at the .05 level with 74 degrees of freedom. Notice in the 1973 study that Q_4 (Relaxed vs. Tense), M (Practical vs. Imaginative), B (Less Intelligent vs. More Intelligent), and Q_1 (Conservative vs. Experimenting) do come close to the .226 figure necessary to be considered significant. This is mentioned here because the computation of the F-ratios did find Q_4, M, Q_1, and I (Tough-minded vs. Tender-minded) as a group to be significant. In the 1975 study with N=140 the .05 level of significance for correlations is about .190. When the grades were weighted as A or B=1 and all others "0", the Q_4 variable with a correlation of .1945 was significant. However, when the grades were weighted A=3, B=2, C=1 and all others =0 the

TABLE 1 (REVISED)

SUMMARY TABLE FOR 16PF SCORE CORRELATIONS
(1973 N=76, 1975 N=140; A=43, B=50, C=29, D*=18)

Variable of 16PF	A&B=1, Other=0 Values*		A=3, B=2 C=1, D=0
	N=76	N=140	N=140
A	-.0058	-.1028	-0.1157
B	.1763	.1328	0.1816
C	-.0484	-.0127	0.0086
E	.0596	.0724	0.0078
F	-.1372	-.0511	-0.0726
G	.0777	-.0902	-0.0228
H	-.0906	-.0856	-0.1209
I	-.1443	-.0966	-0.0958
L	-.0486	.0998	0.0783
M	.1829	.0856	0.0418
N	-.0537	-.1331	-0.0687
O	.1195	.0713	0.0298
Q_1	-.1699	.0132	-0.0230
Q_2	.1051	.1034	0.0863
Q_3	.0101	-.1794	-0.1180
Q_4	.2187	.1945	0.1358

The .05 level of significance is .226 with 74 degrees of freedom and .19 with 138 degrees of freedom.

*The criterion variable values were:
In 1973 and 1975: A and B=1, All Others=0
In 1975 only: A=3, B=2, C=1, All Others=0

TABLE 2

SUMMARY TABLE FOR 16PF REGRESSION ANALYSIS

(1973, N = 76)

Next Variable	Group Correl.	Group Variance	Group F-Ratio	Sign. .05*	Group d/f
Q_4	.2187	.0479	3.719	3.98	1/74
add M	.3162	.1000	4.054	3.13	2/73
" I	.3729	.1391	3.877	2.74	3/72
" Q_1	.4393	.1930	4.244	2.51	4/71
" A	.4666	.2177	3.897	2.35	5/70
" L	.4963	.2463	3.757	2.24	6/69
" E	.5180	.2683	3.562	2.16	7/68
" F	.5348	.2860	3.355	2.09	8/67
" Q_3	.5396	.2912	3.013	2.03	9/66
" H	.5424	.2942	2.710	1.98	10/65
" C	.5478	.3001	2.494	1.94	11/64
" N	.5506	.3032	2.284	1.91	12/63
" B	.5531	.3059	2.102	1.88	13/62
" G	.5538	.3067	1.927	1.86	14/61
" Q_2	.5540	.3069	1.771	1.84	15/60
" O	.5542	.3071	1.634	1.82	16/59

*The .05 level of significance with the corresponding degrees of freedom.

highest correlation was .18 for b (Intelligence). None of these are significant.

Regression Analysis of 16PF Test Variables

A summary of the 16PF score regression analysis is shown in Tables 2, 2a and 2b. The data on each line of the table represents the cumulative effect for the variable and all of the preceeding variables combined into one group. Thus, for example in Table 2, the F-ratio of 4.244 on the line with Q_1 does not represent Q1 alone but the total effect of Q4, M, I, and Q1. Likewise, the group correlation of .4393 and the group variance of .1930 represent the combined effect of the first four variables. Notice in Table 2 that all of the F-ratios except the first one and the last two are above the .05 significance level.

This table (2) indicates that no single variable is a significant predictor, but Q_4 approaches significance. When M is combined with Q_4, the pair becomes significant. Further, when I is added the significant increases and Q_1 is also added, the group of four has an F-ratio (4.244) well above the .05 level of significance (2.51). However, the value of the difference between the F-ratio and the .05 significance level diminishes as each additional variable after Q_1 is included in the group from 1.734 (4.244-2.51) for the first four variables to only .067 (1.927-1.86) after fourteen variables have been included in the group. Then the 16PF test loses its significance completely as the last two variables Q_2 and O are added to the group. However, only the first seven F-ratios of Table 2a and the first five F-ratios of Table 2b are above their corresponding level of significance.

TABLE 2a

SUMMARY TABLE FOR 16PF REGRESSION ANALYSIS
(1975 N=140; A&B=1, Other=0)

Next Variable	Group Correl.	Group Variance	Group F-Ratio	Sign. .05	Group d/f
Q_4	.1945	.0379	5.428	3.89	1/138
Add N	.2309	.0533	3.856	3.03	2/137
" M	.2544	.0647	3.137	2.62	3/136
" B	.2734	.0748	2.727	2.41	4/135
" Q_3	.2964	.0879	2.581	2.25	5/134
" F	.3064	.0939	2.297	2.13	6/133
" C	.3181	.1012	2.123	2.05	7/132
" Q_1	.3236	.1047	1.916	1.98	8/131
" I	.3277	.1074	1.737	1.92	9/130
" O	.3297	.1087	1.573	1.88	10/129
" E	.3325	.1106	1.447	1.8	11/128
" H	.3358	.1128	1.345	1.80	12/127
" Q_2	.3362	.1131	1.236	1.76	13/126
" A	.3364	.1132	1.139	1.72	14/125
(Further Computations Meaningless)					

Note: The first seven groups indicate significance.

Correlations of ACT Scores

The correlations of the ACT scores with the criterion variable of success in EDP training are summarized in Table 3. All of the ACT scores except Natural Science have a correlation above .256, the value required for significance at the .05 level with 58 degrees of freedom.

Regression Analysis of ACT Scores

A summary of the ACT score regression analysis is shown in Table 4. As in Table 2, the data on each line of the table represents the cumulative effect for that variable and all of the preceding variables combined into one group. Thus the data on the line identifying the EN variable represents the total effect of all four of the ACT scores as a group.

Notice that all four of the F-ratios listed were well above the .05 level of significance. The F-ratio of the Mathematics score (10.724) is well above the level of significance (4.00). As each additional ACT score is added to the group, the margin of significance diminishes yet remains above the corresponding significance level.

Negative Prediction Considerations

Since the criterion variable of the 1973 study was dichotomous,

TABLE 2b

SUMMARY TABLE FOR 16PF REGRESSION ANALYSIS
(1975 N=140; A=3, B=2, C=1, Other=0)

Next Variable	Group Correl.	Group Variance	Group F-Ration	Sign. .05	Group d/f
B	.1816	.0330	4.7075	3.89	1/138
Add Q_3	.2255	.0508	3.6688	3.03	2/137
" H	.2486	.0618	2.9851	2.62	3/136
" C	.2688	.0723	2.6290	2.41	4/135
" Q_4	.2785	.0776	2.2541	2.25	5/134
" I	.2867	.0822	1.9849	2.13	6/133
" Q_1	.2927	.0857	1.7666	2.05	7/132
" M	.2967	.0880	1.5805	1.98	8/131
" N	.3001	.0900	1.4292	1.92	9/130
" F	.3029	.0917	1.3028	1.88	10/129
" G	.3048	.0929	1.1916	1.84	11/128
" L	.3062	.0938	1.0952	1.80	12/127
" E	.3069	.0942	1.0079	1.76	13/126
" O	.3077	.0947	.9334	1.72	14/125
" A	.3083	.0950	.8680	1.68	15/124

Note: The first five groups indicate significance.

TABLE 3

SUMMARY TABLE FOR ACT SCORE CORRELATIONS

(1973, N = 60)

Variable of ACT	Correlation with Criterion*	Area of ACT Test Scores
EN	.2607	English
MA	.3950	Mathematics
SS	.2881	Social Studies
NS	.2090	Natural Sciences
CM	.3580	Composite

*The .05 level of significance with 58 degrees of freedom is .256.

TABLE 4

SUMMARY TABLE FOR ACT REGRESSION ANALYSIS

(1973, N = 60)

Next Variable	Group Correl.	Group Variance	Group F-Ratio	Sign. .05*	Group d/f
MA	.3950	.1561	10.724	4.00	1/58
add SS	.4090	.1673	5.726	3.16	2/57
" NS	.4251	.1807	4.117	2.77	3/56
" EN	.4281	.1833	3.086	2.54	4/55

*The .05 level of significance with the corresponding degrees of freedom.

the variables which were found to be the best predictors of success were also the best predictors of lack of success. That is, a person lacking in mathematical skills or with personality trait scores in the opposite direction from scores indicating probable success, would more likely be unsuccessful. Students with higher 16PF Q4 (Relaxed vs. Tense) and M (Practical vs. Imaginative) scores coupled with lower I (Tough-minded vs. Tender-minded) and Q_1 (Conservative vs. Experimenting) scores were found to be more successful while students with lower Q_4 and M scores coupled with higher I and Q_1 scores were found to be less successful. No attempt was made in the 1975 study to identify negative predictors as the same logic should apply. That is, the lack of a quality identified as a predictor of success should indicate that success is less probable.

Summary of Findings - 1973

These findings were based on the four tables presented earlier in this chapter which summarize the results of the statistical procedure. They included the correlation tables and regression analyses for both the 16PF scores and the ACT scores.

None of the 16PF variables taken individually had a sufficiently high correlation with the criterion, either positive or negative, to be considered significant at the .05 level. The four highest are Q_4 (Relaxed vs. Tense), M (Practical vs. Imaginative), B (Less Intelligent vs. More Intelligent), and Q_1 (Conservative vs. Experimenting) which have .2187, .1829, .1763, and-.1699 correlation coefficients respectively. The .05 level with 74 degrees of freedom is .226, and is greater than any found in Table 1.

The regression analysis also indicated, as did the correlations, that when any single 16PF variable was considered individually it was not significant since the largest F-ratio for a single variable, Q_4, was only 3.719. An F-ratio of 3.98 would be needed to be significant at the .05 level.

However, when two or more variables were considered as a group, each of the groups was found to be significant except the last two, those containing Q2 (Undisciplined Self-conflict vs. Controlled) and O (Self-assured vs. Apprehensive). The highest F-ratio in Table 2 was 4.244 for the group of four variables Q4 (Relaxed vs. Tense), M (Practical vs. Imaginative), I (Tough-minded vs. Tender-minded), and Q_1 (Conservative vs. Experimenting). Thus this group of four variables was the most significant predictor of the criterion variable that could have been selected from any combination of the 16PF predictor variables. Further, it was well above the 2.51 figure necessary to be significant at the .05 level.

All but one of the ACT score variables were found to correlate sufficiently with the criterion to be considered significant. The .05 level of significance with 58 degrees of freedom was .256 and, as shown in Table 3, only the Natural Sciences (NS) correlation (.209) was below that level. Mathematics had the highest correlation coefficient at .3950 with the composite score next at .3580. Social Studies and English followed with .2881 and .2607 correlation coefficients respectively.

The regression analysis for the ACT scores, as shown in Table 4, indicated that each of the F-ratios was well above its corresponding .05 level of significance. The Mathematics F-ratio was 10.724 with 4.00 considered significant. Social Studies and Mathematics combined had an F-ratio of 5.726 with 3.16 being the value of the .05 level of significance. When Natural Sciences was combined with the first two, the F-ratio was 4.117 and 2.77 was the significance level. When all four ACT scores were considered as one group the F-ratio was 3.086 while 2.54 was the significance level. Thus Mathematics alone was a better predictor than any combination of ACT scores.

In relating the findings to the hypotheses, the correlation data for the 16PF scores, indicate that none of the 16PF scores when considered singularly correlated sufficiently with the criterion variable to be significant at the .05 level. However, data from the stepwise regression analysis indicated that several selected combinations of the 16PF scores did have F-ratios above the .05 level of significance. Thus the first null hypothesis was separated into two parts:

1. 16PF scores considered individually, which was accepted, and
2. 16PF scores considered in selected combinations, which was rejected.

The second part of the first hypothesis was rejected because although none of the individual 16PF scores were significant, three combinations (1) Q4 and M, (2) Q4, M and I, and (3) Q4, M, I, and Q_1 all

had F-ratios well above the .05 level of significance.

The F-ratio of 4.244 for the first four variables, Q_4 (Relaxed vs. Tense), M (Practical vs. Imaginative), I (Tough-minded vs. Tender-minded), and Q_1 (Conservative vs. Experimenting), was the highest in the study and indicated that this group of four, when used as a group and not individually, was the best predictor of the criterion variable to be found in the 16PF test. Although all of the other variables were progressively added to this group of four and the F-ratio still remained above the .05 significance level, except for the last two, none of these other groups was as significant as the group of the first four variables.

The first part of the second null hypothesis was rejected because three of the ACT scores could be used individually to predict a student's success in the EDP course.

The regression analysis likewise supported the rejection of the second part of this hypothesis on the basis of the F-ratios. All of the group F-ratios were above their significance level, but the significance of the F-ratio was considerably reduced with the inclusion of each additional ACT score.

Both the correlations and the F-ratios of the Mathematics and Social Studies ACT scores indicated a firm rejection of the second hypothesis for these two variables. The other two variables, English and Natural Science, had little predictive value.

Summary of Findings - 1975

These findings are based on Table 1 (Revised) and Tables 2, 2a and 2b. The 16PF variable Q_4 (Relaxed vs. Tense) correlated at .1945 which is significant at the .05 level when only two levels of success were indicated, while Q_3 (Undisciplined vs. Controlled) is almost significant with a negative .1794. However, when success was indicated in four levels, this significance was lost and the variable B (Intelligence) correlation increased to .1816 slightly below the .19 significance level.

The regression analysis using a dichotomous criterian variable also indicated that Q_4, with an F-ratio of 5.428 was well above the .05 significance level of 3.89. However, when four levels of success were indicated, variable B, with an F-ratio of 4.70, was well above the .05 significance level.

In relating the 1975 findings to the first hypothesis, the hypothesis was totally rejected. The first part was rejected in each regression analysis, but because of a different variable; Q_4 in the dichotomous analysis and B in the four level analysis. The second part was also rejected by each analysis but by different combinations of variables; seven (Q_4, N, M, B, Q_3, F and C) in the dichotomous analysis and five (B, Q_3, H, C, and Q_4) in the four level analysis. Notice that B, C, Q_3 and Q_4 appear in both combinations.

CHAPTER IV - CONCLUSIONS AND RECOMMENDATIONS

Conclusions - 1973

The conclusion drawn from the 1973 testing of the first hypothesis was that none of the 16PF scores could be used singularly as a predictor of success in EDP, but when the four variables Q_4 (Relaxed vs. Tense), M (Practical vs. Imaginative), I (Tough-minded vs. Tender-minded), and Q_1 (Conservative vs. Experimenting) were considered collectively as a selected group, they were considered significant predictors.

The conclusion drawn from the testing of the second hypothesis was that the ACT Mathematics, Social Studies, English, and Composite scores could all be used as predictors of success in EDP, but Mathematics alone was the best single predictor.

In summary, the more successful students in this study possessed the following characteristics which were considered valid predictors of success in EDP training:

1. Above average ACT Mathematics scores; 2. Above average ACT Composite scores; 3. Above average ACT Social Studies scores; 4. Above average ACT English scores; 5. Above average 16PF Q_4 scores indicating more Tense; 6. Above average 16PF M scores indicating more Imaginative; 7. Below average 16PF I scores indicating more Tough-minded; 8. Below average 16PF Q_1 scores indicating more Conservative.

Thus the students with good mathematics and communication skills who were more tense, imaginative, tough-minded, and conservative had the best record of success in EDP training.

Conclusions - 1975

The conclusions drawn from the 1975 testing of the first hypothesis were that two of the 16PF scores could be used as singular prodictors of success in EDP. The personality factor of Q_4 (Relaxed vs. Tense) can predict success when only two levels of success are indicated but is not a good predictor for four levels of success. The factor B (Intelligence) is a good predictor when four levels of success are indicated but not when only two levels of success are indicated.

In summary, the results of the three analyses of the 16PF scores are not consistant with each other. However, there is sufficient evidence in these studies to indicate that the 16PF inventory, or at least parts of it, can be used as a predictor of success in EDP training.

Recommendations

This study was based on the premise that certain personality characteristics would tend to make students either successful or unsuccessful in EDP training courses. The purpose of this study was to identify any personality traits, using the 16PF inventory and ACT scores, that migh have been possessed by either successful or unsuccessful students.

The findings of these studies did support the premise with several of the variables. This indicates that there is considerable validity to the premise and that more research needs to be done in this field.

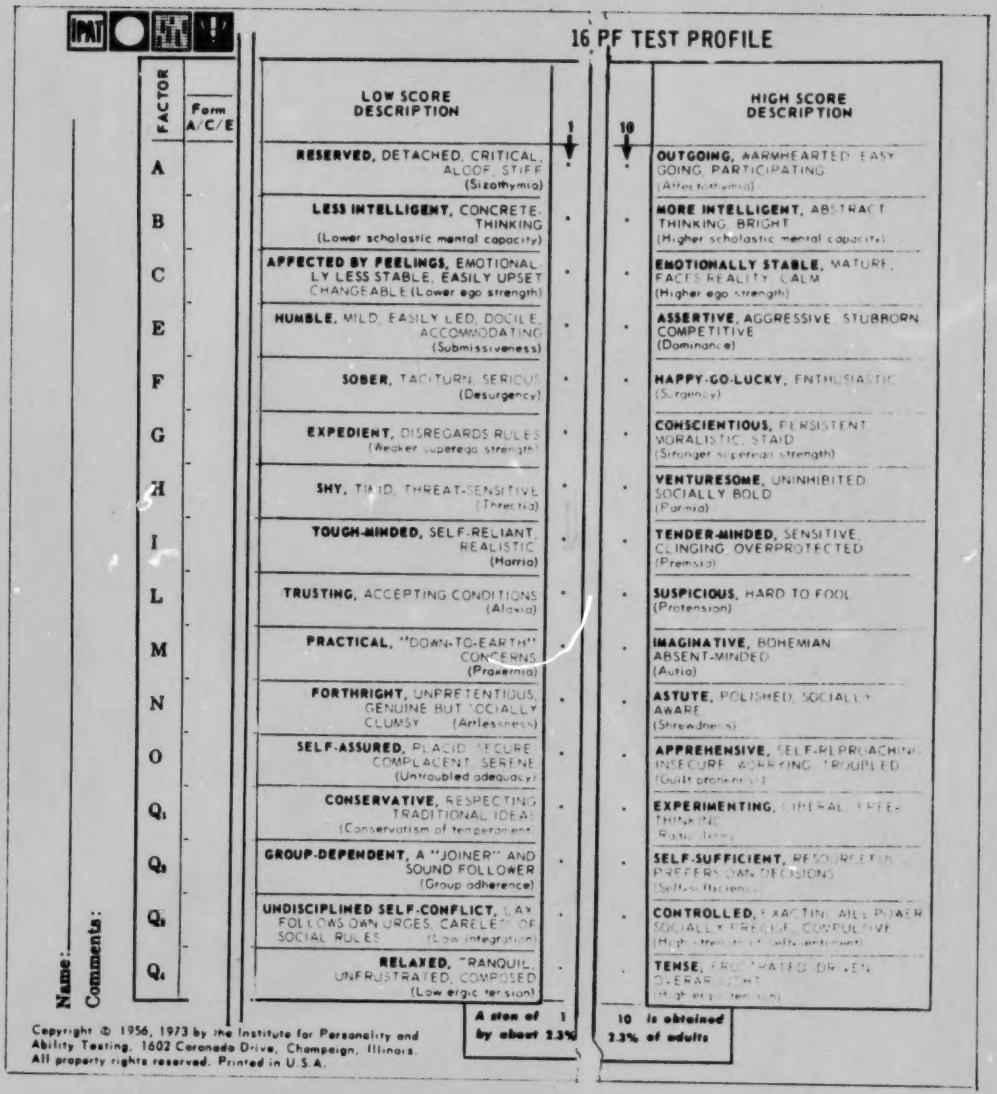
Thus the following recommendations for further study are presented: (1) A replication of this study be made with a much larger number of subjects; (2) A similar sutdy by designed using a different set or combination of instruments; (3) A thorough investigatior of individual personality factors as predictors of success in any type of problem solving skills.

APPENDIX A

GLOSSARY OF ABBREVIATIONS

ACB	Army Classification Battery
ACT	American College Test
ATPP	Aptitude Test for Programming Personnel
CPAB	Computer Programming Aptitude Battery
CQT	College Qualification Test
EDP	Electronic Data Processing
EPPS	Edwards Personality Preference Scale
GPA	Grade Point Average
HSPQ	High School Personality Questionnaire
IBM	International Business Machines Corporation
MIQ	Minnesota Importance Questionnaire
MMY	Mental Measurements Yearbook
MVII	Minnesota Vocational Interest Inventory
PAT	Programmers Aptitude Test
16PF	Sixteen Personality Factor Questionnaire
SPSS	Statistical Package for the Social Sciences
SRA	Science Research Associates

APPENDIX B


16 PF TEST PROFILE
FACTOR
Form A/C/E
LOW SCORE DESCRIPTION
HIGH SCORE DESCRIPTION
1
10
A
RESERVED, DETACHED, CRITICAL, ALOOF, STIFF (Sizothymia)
OUTGOING, WARMHEARTED, EASY GOING, PARTICIPATING (Affectothymia)
B
LESS INTELLIGENT, CONCRETE-THINKING (Lower scholastic mental capacity)
MORE INTELLIGENT, ABSTRACT THINKING, BRIGHT (Higher scholastic mental capacity)
C
AFFECTED BY FEELINGS, EMOTIONALLY LESS STABLE, EASILY UPSET CHANGEABLE (Lower ego strength)
EMOTIONALLY STABLE, MATURE, FACES REALITY, CALM (Higher ego strength)
E
HUMBLE, MILD, EASILY LED, DOCILE, ACCOMMODATING (Submissiveness)
ASSERTIVE, AGGRESSIVE, STUBBORN COMPETITIVE (Dominance)
F
SOBER, TACITURN, SERIOUS (Desurgency)
HAPPY-GO-LUCKY, ENTHUSIASTIC (Surgency)
G
EXPEDIENT, DISREGARDS RULES (Weaker superego strength)
CONSCIENTIOUS, PERSISTENT, MORALISTIC, STAID (Stronger superego strength)
H
SHY, TIMID, THREAT-SENSITIVE (Threctia)
VENTURESOME, UNINHIBITED, SOCIALLY BOLD (Parmia)
I
TOUGH-MINDED, SELF-RELIANT, REALISTIC (Harria)
TENDER-MINDED, SENSITIVE, CLINGING, OVERPROTECTED (Premsia)
L
TRUSTING, ACCEPTING CONDITIONS (Alaxia)
SUSPICIOUS, HARD TO FOOL (Protension)
M
PRACTICAL, "DOWN-TO-EARTH" CONCERNS (Praxernia)
IMAGINATIVE, BOHEMIAN, ABSENT-MINDED (Autia)
N
FORTHRIGHT, UNPRETENTIOUS, GENUINE BUT SOCIALLY CLUMSY (Artlessness)
ASTUTE, POLISHED, SOCIALLY AWARE (Shrewdness)
O
SELF-ASSURED, PLACID, SECURE, COMPLACENT, SERENE (Untroubled adequacy)
APPREHENSIVE, SELF-REPROACHING, INSECURE, WORRYING, TROUBLED (Guilt proneness)
Q1
CONSERVATIVE, RESPECTING TRADITIONAL IDEAS (Conservatism of temperament)
EXPERIMENTING, LIBERAL, FREE-THINKING (Radicalism)
Q2
GROUP-DEPENDENT, A "JOINER" AND SOUND FOLLOWER (Group adherence)
SELF-SUFFICIENT, RESOURCEFUL, PREFERS OWN DECISIONS (Self-sufficiency)
Q3
UNDISCIPLINED SELF-CONFLICT, LAX, FOLLOWS OWN URGES, CARELESS OF SOCIAL RULES (Low integration)
CONTROLLED, EXACTING WILL POWER, SOCIALLY PRECISE, COMPULSIVE (High integration, will control)
Q4
RELAXED, TRANQUIL, UNFRUSTRATED, COMPOSED (Low ergic tension)
TENSE, FRUSTRATED, DRIVEN, OVERWROUGHT (High ergic tension)
Name:
Comments:
A sten of 1 is obtained by about 2.3% of adults
10 is obtained 2.3% of adults
Copyright © 1956, 1973 by the Institute for Personality and Ability Testing, 1602 Coronado Drive, Champaign, Illinois. All property rights reserved. Printed in U.S.A.

APPENDIX C

DISTRIBUTION OF 16PF SCORES

N=76 (50 Successful/26 Unsuccessful)

Each entry represents the number of students in each 16PF score range. Successful vs. Unsuccessful. Scores were grouped for brevity.

Sixteen Personality Factor Scores (Grouped for Brevity)					
Sten Scores Factors	1 - 2 S/U	3 - 4 S/U	5 - 6 S/U	7 - 8 S/U	9 - 10 S/U
A	6/3	25/12	13/ 8	6/ 2	0/1
B	0/4	13/ 6	19/ 9	16/ 5	2/2
C	2/3	13/ 5	26/10	6/ 4	3/4
E	2/2	9/ 8	29/ 8	8/ 8	2/0
F	7/2	10/ 5	22/12	9/ 4	2/3
G	2/1	8/ 5	18/10	15/10	7/0
H	0/2	22/ 5	15/ 9	11/ 8	2/2
I	7/0	9/ 8	23/11	9/ 5	2/2
L	0/0	13/ 9	24/ 8	11/ 6	2/3
M	8/4	12/14	21/ 5	7/ 3	2/0
N	0/1	7/ 5	21/ 5	17/ 8	5/7
O	4/2	6/ 5	17/10	18/ 7	5/2
Q1	0/0	11/ 5	23/ 7	15/12	1/2
Q2	2/2	12/ 5	16/12	12/ 7	8/0
Q3	2/0	4/ 3	16/12	22/ 6	6/5
Q4	1/4	10/ 7	21/ 8	11/ 6	7/1

APPENDIX D

DISTRIBUTION OF ACT SCORES

N=60 (39 Successful/21 Unsuccessful)

Each entry indicates the number of successful vs. total students and the percent of succesful students in each ACT score group.

ACT Score Groups	0 - 10	11 - 15	16 - 20	21 - 25	26+
English S/T	4/9	6/9	11/20	17/21	1/ 1
Percent	44	67	55	81	100
Math. S/T	0/3	3/7	9/15	16/20	11/15
Percent	0	43	60	75	73
Social Stud. S/T	3/4	4/8	6/15	14/18	12/15
Percent	75	50	40	78	80
Nat. Sci. S/T	1/5	6/12	12/15	13/20	7/ 8
Percent	20	50	80	65	88
Composite S/T	1/2	2/8	17/26	13/18	6/ 6
Percent	50	25	65	72	100

Note that several of the score groups are quite small, thus the percentage of success may be misleading

APPENDIX E

PROFILE OF 16PF INPUT DATA

N=76 (50 Successful/26 Unsuccessful)

16PF Factor	Mean N=76	Stnd. Dev. N=76	Stnd. Devs. N=50/26	Means N=50/26	Mean Diff. S - U
A	4.41	1.87	1.76/2.12	4.40/4.42	- .02
B	5.50	1.87	1.69/2.14	5.74/5.04	.70
C	5.33	1.98	1.83/2.22	5.26/5.46	- .20
E	5.45	1.69	1.59/1.86	5.52/5.31	.21
F	5.25	2.12	2.06/2.18	5.04/5.65	- .61
G	5.97	1.90	2.01/1.65	6.08/5.77	.31
H	5.46	1.85	1.68/2.11	5.34/5.69	- .35
I	5.22	1.96	2.10/1.57	5.02/5.62	- .60
L	5.64	1.85	1.67/2.14	5.58/5.77	- .19
M	4.43	2.02	2.15/1.62	4.70/3.92	.78
N	6.43	1.91	1.67/2.31	6.36/6.58	- .22
O	5.89	2.15	2.15/2.10	6.08/5.54	.54
Q1	5.95	1.53	1.48/1.56	5.76/6.31	- .55
Q2	5.82	1.90	2.03/1.60	5.96/5.54	.58
Q3	6.53	1.88	1.85/1.95	6.54/6.50	.04
Q4	5.68	2.13	2.00/2.21	6.02/5.04	.98

APPENDIX F

PROFILE OF ACT INPUT DATA

(39 Successful/21 Unsuccessful)

ACT Area	Mean N=60	Stnd. Dev. N=60	Stnd. Devs. N=39/21	Means N=39/21	Mean Diff. N=39/21
EN	17.72	4.97	4.51/5.28	18.67/15.95	2.72
MA	20.88	5.98	4.84/6.53	22.62/17.67	4.95
SS	18.87	5.73	5.06/6.20	20.08/16.62	3.46
NS	20.47	6.32	6.62/5.26	21.44/18.67	2.77
CM	19.58	4.71	3.97/5.10	20.82/17.29	3.53

EN for English
MA for Mathematics
SS for Social Studies
NS for Natural Science
CM for Composite

APPENDIX G

PROFILE OF 16PF MEANS
(N=140, 93 Successful/47 Unsuccessful)

16PF Factor	N 140	S 93	U 47	Mean Diff S-U
A	4.19	4.07	4.43	-.36
B	5.01	5.20	4.64	.56*
C	5.33	5.31	5.36	-.05
E	5.96	6.05	5.77	.28
F	5.47	5.40	5.62	-.22
G	5.59	5.47	5.83	-.36
H	5.62	5.50	5.85	-.35
I	4.35	4.23	4.60	-.37
L	5.86	6.00	5.60	.40
M	4.99	5.10	4.77	.33
N	5.35	5.15	5.74	-.59*
O	5.32	5.42	5.13	.29
Q_1	5.95	5.97	5.91	.06
Q_2	6.00	6.14	5.72	.42
Q_3	5.71	5.47	6.19	-.72
Q_4	5.70	5.96	5.19	.77*

*Note that the difference between the means is greater than half a point for B (Intelligence), N (Shrewdness) and Q_4 (Tenseness). The more intelligent, less shrewd and more tense were more successful.

APPENDIX H

PROFILE OF 16PF MEANS
(N=140, A=43, B=50, C-29, D*=18)

16PF Factor	N 140	A 43	B 50	C 29	D* 18	Diff A-D
A	4.19	3.88	4.22	4.48	4.33	-.45
B	5.01	5.40	5.04	5.00	4.06	1.34
C	5.33	5.47	5.18	5.31	5.44	.03
E	5.96	5.81	6.26	5.62	6.00	-.19
F	5.47	4.98	5.76	5.97	5.06	-.08
G	5.59	5.67	5.30	6.03	5.50	.17
H	5.62	5.19	5.78	5.90	5.78	-.59
I	4.35	4.09	4.34	4.69	4.44	-.35
L	5.86	6.02	5.98	5.48	5.78	.24
M	4.99	5.04	5.14	4.55	5.11	-.07
N	5.35	5.51	4.84	5.69	5.83	-.32
O	5.32	5.40	5.44	4.86	5.56	-.16
Q_1	5.95	5.84	6.08	5.79	6.11	-.27
Q_2	6.00	6.44	5.88	5.24	6.50	-.06
Q_3	5.71	5.51	5.44	6.48	5.72	-.21
Q_4	5.70	5.93	5.98	4.93	5.61	.32

*Includes all except A, B and C grades.

Note that the differences between the means of A and D are all less than half a point except B (Intelligence) and H (Shyness). The more intelligent and more shy were more successful.

EGOLESS PROGRAMMING: AS AN INSTRUCTIONAL TECHNIQUE

Paul Cheney
College of St. Thomas
St. Paul, Minnesota

ABSTRACT

This study attempted to find out if students could learn a computer programming language, to a greater degree in a given amount of time, if they wrote their programs in pairs rather than individually. The results of this research suggests that the paired students did learn more programming concepts than those who worked individually.

INTRODUCTION

This paper describes a small research project that was conducted at the College of St. Thomas in St. Paul, Minnesota. The research was undertaken in an attempt to find a better way of teaching an introductory class how to program in a short period of time. Most of the Quantitative Methods classes at St. Thomas involve a substantial amount of statistics and therefore the programming must be taught and grasped quickly so that it can be utilized in the remainder of the course. From my own experience and that of my colleagues, I was encouraged to study the possibility that students might gain a better understanding of computer programming if they programmed in pairs rather than individually. This was the principle hypothesis I attempted to investigate.

EGOLESS PROGRAMMING

Egoless programming is a term coined by Gerald Weinberg in his book The Psychology of Computer Programming [1]. In essence it involves removing the problem of the programmer's ego from the programming process by restructuring the environment in which programs are written. Many instructors, have in the past, emphasized individuality because this seemed to be a necessary prerequisite to adequately measuring student performance. Egoless programming in contrast involves the process of one's colleagues inspecting his work for errors prior to the start of computer testing. The objective being to improve programmer productivity.

In the classroom, productivity is measured in terms of the degree of programming ability that can be achieved in a specified time period. The objective in the experiment described herein, was to examine the proposition that the egoless programming structure would increase programming productivity in the classroom sense.

In an industrial setting many programmers are often secretive and unwilling to share much of their knowledge with their colleagues. There are two principle reasons for this:

1. Programmers are usually evaluated on an individual rather than on a group basis. A programmer who carefully guards his trade secrets may receive a better evaluation due to the fact that his project deadlines are more often met and his project budgets are seldom exceeded.
2. Programmers are hesitant to let others inspect their programs because they are afraid that others will discover their inadequacies.

When this type of environment exists, everyone loses; the company loses because the programmers are not as efficient as they could be and the programmers lose because they are forced to work under additional anxiety until they find problem solutions on their own.

A similar situation exists in the classroom. Students are often faced with time deadlines, just as industrial programmers are, and they are in many cases assessed penalties for not meeting those deadlines. Many students are secretive about their programming knowledge because they realize that one way they can gain an advanced standing in the class is to meet more of their deadlines than their classmates. Students also are often reluctant to let others see their programs for fear they will be embarrassed by their mistakes. In addition, many instructors stress that programs must be done individually. Collaboration could be viewed as cheating, and the determination of how much help one student can legitimately give another is not an easy task for students or instructors. The result is an environment in which most learning must be accomplished on an individual basis. This is a restrictive and undesirable situation that inhibits the degree of learning that can take place. Harold Sackman [2] conducted a recent study involving trainees at the Air Force Academy. One of his observations was that the restriction to individual work, which is particularly emphasized at the Academy, seemed to adversely affect learning. He stated that:

> This resulted because group cooperation, which is essential in any real programming effort, was set aside by the school as "cheating."

I agree with Weinberg when he said:

> Average people can be trained to accept their humanity - their inability to function as a machine - and to value it and work with others so as to keep it under the kind of control needed if programming is to be successful.

I believe that it is time for programming instructors to provide the necessary environment within which the maximum amount of learning can take place.

EXPERIMENTAL DESIGN

The sample consisted of the students enrolled in one of my introductory courses at the College of St. Thomas. There were a total of 42 students in the class and each student was randomly placed into one of two test groups. The students in the first test group were given specific instructions to work completely alone (i.e. see Exhibit 1). Consultants were instructed to answer only specific questions about program instructions or in some cases, program segments. The students in the second test group were separated into pairs. They were told that the programs they wrote were products of the pair not of an individual. Each member of the pair was told to check his partner's code for possible errors prior to any computer testing.

Both groups were assigned the same set of six programs. Whether they were working in pairs or individually each student was initially to code his own version of the program. Once this was done each individual in group two (i.e. paired group) exchanged programs with his partner and checked the code for possible errors. This procedure would be repeated until no more errors could be found. It was only at this time that computer testing began. Each individual in the class was required to turn in the following documentation:

- flowchart
- description of the input and output
- original version of the program code
- computer printout

In addition, those students who were programming in pairs were required to hand in a list of the errors found, who found them, and during which iteration. It was felt that this would encourage these students to actually check over their partner's code for errors.

The programs were assigned a relatively low value in terms of the grading for the course (i.e. approximately 5%). It was stressed, however, that doing the programs would be a learning experience that would be valuable when taking the programming exam. The programming exam was composed of a set of objective questions and two programs. It represented approximately 25% of the total number of points in the course.

Programming ability was operationalized in terms of the programming exam scores. All the exams were graded by the same person on the same day using the same criteria.

STUDY RESULTS

The results of the study are presented in Exhibit 2. With sample sizes less than 30 I chose to do a t-test of a hypothesis concerning the difference between the mean exam scores for the two groups. At a level of significance of .01 the group that programmed in pairs did do significantly better on the programming exam than those who programmed individually (see Exhibit 2).

Exhibit 1

Instructions Given to Students in Both Programming Groups

The class will be divided into two groups and each will write their programs in a slightly different manner.

Group 1

> Those in this group will write their programs completely alone without any help from their fellow students. They may seek aid from the computer consultants or the instructor.

Group 2

> Those in this group will work alone initially until the first version of the program has been coded on paper. Each person in this group will be assigned a partner and when both partners have finished writing their own first version of the program, they will exchange programs and check the other person's program for errors. Any errors will be communicated to the original party and the process will be repeated until both partners are satisfied that the programs are error free. Then computer testing of the programs will begin and continue until one or both of the programs are in working order. The two partners may communicate with each other as much as they desire. Either partner may also seek aid from the computer consultants or the instructor.
>
> The pair may hand in both programs, but one program in working order will be sufficient to get full credit. Each program will be evaluated and both members will receive the same grade.

Exhibit 2

Null Hypothesis: $m_1 - m_2 = 0$

Alternative Hypothesis: $m_1 - m_2 > 0$

m_1 = Individual population mean

m_2 = Paired population mean

Group 1 - Individuals

Sample Mean	$\bar{X}_1$	= 74.4
Sample Variance	s_1^2	= 159.3
Sample Std. Dev.	s_1	= 12.6
Sample Size	n_1	= 20

Group 2 - Pairs

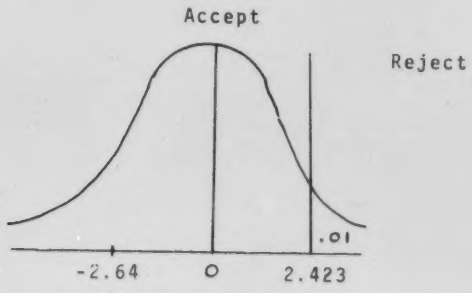
Sample Mean	$\bar{X}_2$	= 83.9
Sample Variance	s_2^2	= 117.07
Sample Std. Dev.	s_2	= 10.8
Sample Size	n_1	= 22

$$t = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}} \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}}$$

$$t = \frac{74.4 - 83.9}{\sqrt{\frac{(19)(159.3) + (21)(117.07)}{40}} \sqrt{\frac{1}{20} + \frac{1}{22}}}$$

$t = -2.64$

The degrees of freedom was equal to 40 (i.e. $n_1 + n_2 - 2$) and the dividing line between accepting and rejecting my one-sided hypothesis was 2.423. My value for t was-2.64 Therefore the hypothesis can be accepted. The mean exam score for those who programmed individually was significantly less than the mean exam score for those who programmed in pairs.



CONCLUSIONS

The advantages of using the egoless structure in industry is evident. Weinberg cites several actual advantages. [3] They include:

1) program specifications are more often met
2) variations in debugging time is reduced therefore estimates are more realistic
3) the adaptability of programs is improved because at least two people are capable of understanding the program
4) the reading of programs is a good training technique.

I believe that in the classroom we can achieve all of these advantages and in addition I feel we can say that it is certainly possible to expand upon Weinberg's point number 4 and say that: "productivity in the classroom can be improved if the egoless structure is utilized."

References

1 Weinberg, G. M., The Psychology of Computer Programming, New York, Van Nostrand Reinhold, 1971.

2 Sackman, Harold, Man-Computer Problem Solving: Experimental Evaluation of Time-Sharing and Batch Processing, N.J., Auerbach, 1970

3 Weinberg, G. M., The Psychology of Improved Programming Performance, Datamation, November, 1972, pg. 82-85.

DESCRIPTION AND EVALUATION OF THE WEBSTER'S DIACRITICAL MARKINGS COMPUTER - ASSISTED INSTRUCTIONAL PROGRAM.

Prepared by

James R. von Feldt, D.M.A.
Joanne Subtelny, Ph.D.

National Technical Institute for the Deaf
Rochester Institute of Technology
Rochester, New York

Rationale for Program. The English language presents major difficulties for deaf students in both the written and spoken form. The rules of grammar are difficult to learn and apply. Spellings often provide little or no clues to the pronunciation of words. In introducing new words with their appropriate pronunciation and spelling to students, NTID classroom teachers have used various methods. Some have used the International Phonetic Alphabet, the Northampton Symbols, or home-made symbol systems. The use of one system throughout NTID would indeed be less confusing and very beneficial. Webster's diacritical system was selected for use in training NTID students and faculty because it will be most accessible in the student's future job situation.

The need to adopt and use a uniform system in presenting new words in writing as they sound is based upon the concept that a central problem in post-secondary education of the deaf is language deficiency. This deficiency is expressed in many ways -- only one of which is difficulty in achieving intelligible speech.

The Webster diacritical system provides a discrete symbol for each sound and designates the appropriate syllable to be stressed in any polysyllabic word. The symbol system, therefore, presents cues for correct production, auditory discrimination and visual recognition of new words in print and as visible speech gestures.

Since 44% of the students entering NTID in the summer of 1975, had intelligible speech, and 37% had sufficient hearing to discriminate some words within sentence context, it was apparent that new words should be presented auditorily, as they appear in print, and in a graphic system offering additional cues for correct pronunciation. Classroom instructors who follow the established recommended procedure of presenting new words in print, in diacritical symbols, in spoken form and within meaningful context, accomplished a great deal in reducing the language deficiency of deaf students. This procedure reinforces reading, spelling, auditory, speech and speech recognition skills. Whereas the majority of students cannot be expected to pronounce the word appropriately or to hear it accurately enough for identification, the multi-sensory approach in introducing words does make sense for all students. Those who cannot auditorily and/or visually perceive the word as spoken, can at least develop recognition and comprehension of that word in printed form.

In order to incorporate a multi-sensory approach in teaching new words to NTID students, it is mandatory that faculty, as well as students, develop competence in the use of diacritical symbols.

To determine the competence of new staff members, 14 new staff members were given a short test (Summer, 1973) to evaluate their knowledge and functional usage of the diacritical symbols. These results revealed none of the new staff (0%) achieved a passing grade of 80%, considered to indicate adequate knowledge of the system. These results clearly indicate that new staff, at the time of employment, do not possess requisite skills, thereby demonstrating the need for an instructional program for faculty. The students were also tested and found to need instruction.

Since all NTID faculty are attempting to communicate information, it is advisable for members of the faculty to impact in a uniform, multi-sensory manner, thus reinforcing each other in a combined effort to upgrade the language competence of all NTID students.

Use of a common diacritical system was strongly recommended as one way of accomplishing this purpose. To implement this recommendation, a computer-assisted instructional package with audio was proposed to develop faculty and student skill in writing and reading words as they sound.

To summarize, the theoritical basis for using the diacritical symbols was established; 1) to facilitate teaching and learning new words, and 2) to improve communication skills of NTID students and foster competence in independent use of the dictionary. The need to develop skill in use of the system among new staff members and among NTID students has been documented. If the system is to be effective, it is obvious that both teachers and students require functional knowledge of its use.

Objective. The objective of the Webster's Diacritical CAI Program is to provide skills in recognizing and using selected diacritical markings for 2 populations:

1. Newly employed faculty and staff at NTID, and

2. NTID students who have some auditory discrimination for speech.

The theoretical basis for this activity is to improve classroom instruction and to reinforce communication skills of students and faculty at NTID.

Description of the Program. The instructional package utilizing computer assisted instruction with audio-visual cues was developed with content purposely restricted in scope to remain within a one to two hour time limit and to meet differing levels of skill at entry. The content of the program was organized in three parts to include all symbols for the vowels, diphthongs and consonants c, ng, x, and qu. Sequential frames demonstrating the instructional strategy are presented in Appendix 1. A flowchart designating the order of presentation appears in Appendix 2. (Appendicies are not included in this paper. They are available from the authors upon request.)

Photographs of a speaker producing the vowels and diphthongs were assembled in a handbook to use while viewing the program. The CAI portion of the program was developed for an IBM 1130 - 1500 system using cathode-ray tube and random access audio unit. Audio messages consisting of the sounds and words used in the program were recorded and appropriately inserted in the program. In addition, a pre-post test (Appendix 3) and three review quizzes were constructed.

The program was debugged and corrected Fall Quarter, 1973, after a pilot investigation involving 15 faculty members in the Division of Communication Programs and 4 NTID students. The final revision made Spring, 1974, increased the passing criterion for the program from 65% to 80%. Data accumulated Summer 1974 from 15 new staff members showed the final revision was definitely beneficial. This revision required more drill and practice before a participant was permitted to progress in the program. The beneficial effect of the revision was documented by the fact that the post-test means showed a 9% increase over previous means attained before increasing the passing criterion. Pre-test scores consistenly have documented the need to train new staff. The mean pre-test score for 24 faculty enrolled in new staff training Summer, 1974, was 39.8%.

Evaluation of the Program. Pre-post test data has been accumulated for 40 new staff members and for 55 NTID students who have taken the Webster's Diacritical CAI Program. As reported in Table 1, the pre-test mean for faculty was 49.35%, sd 17.91, with a post-test mean of 91.48%, sd 6.99. The t value (-16.243) indicates the gain was significant beyond the .01 level of confidence. The 91% post-test mean also indicates that all faculty met our established criterion of 80% after completing the program. None of the faculty scored lower than 80%. Individual pre- and post-tests scores for faculty are graphed in Figure 1.

Table 1. Means, standard deviations and t values for test scores before and after taking the Webster's Diacritical Markings Computer Assisted Instruction Program.

Group	n	Pre-test		Post-test		t
NTID students	55	mean	54.39%	mean	86.14%	-21.146%**
		sd	13.28	sd	8.32	
NTID Faculty	40	mean	49.35%	mean	91.48%	-16.243%**
		sd	17.91	sd	6.99	

Table 1 also includes pre-post test data for the students who have completed the Webster's Diacritical CAI Program. The pre-test mean for the 55 students was 54.39%, sd 13.28, with a post-test mean of 86.14%, sd 8.32. The t value (-21.146) again indicates a significant difference in pre- and post-test performance. Eighty-eight percent of the students achieved the criterion of 80% or better in the post-test situation. These results are graphed in Figure 2.

The comparison between faculty and student groups are summarized in Table 2. As shown, faculty scored slightly lower (mean 49.35%) than students (mean 54.39%) in pre-testing.

Table 2. Means, standard deviations and t values for pre- and post-test scores on the Webster's Diacritical Test for faculty and student groups.

	GROUP				
	Faculty (n=40)		Student (n=55)		t
Pre test	mean	49.35%	mean	54.39%	1.577
	sd	17.91	sd	13.28	
Post test	mean	91.48%	mean	86.14%	-3.299**
	sd	6.99	sd	8.32	

This difference was not statistically significant (+ 1.577) and may be attributed to the fact that the students had received more orientation to the diacritical system before taking the course. The faculty received only a brief one hour orientation, explaining the rationale for learning Webster's diacritical markings before beginning the program.

Post-test results for faculty (mean 91.48%, sd 6.99) and students (mean 86.14%, sd 8.32) revealed the faculty post-test performance was significantly better than the hearing impaired student group (t -3.299).

In both groups, standard deviations for post-tests as compared to pre-tests were greatly reduced. To illustrate, in the faculty group, pre-test scores ranged from 13% correct to 77% correct; i.e., indicating a range in scores of 64%. Post-test scores of the faculty show a restricted range of 20 percent (80% to 100% correct). This observation applied, to a lesser extent, in the student data as reported in Table 2. The pre-test - post-test reduction of standard deviations (faculty from 17.91 to 6.99, and student from 13.28 to 8.32) indicated that both groups expressed rather homogenious post-test performance. Although entering the program with varying levels of skills, both student and faculty groups exited the program with a similar level of skill. This is interpreted as one indication that the CAI program was effective regardless of individual entry level. This finding is further supported by the significant correlations between gain and pre-test scores for both groups, as reported in Table 4. Pre- and post-tests means with gain are graphically displayed for both groups in Figure 3.

In Table 3, percentage gain and time on line is presented for both groups. The mean for faculty gain (42.15%) was significantly greater than the mean for students (31.92%), however, there was no significant difference relative to time on line. On the average, both groups completed the program in less than an hour and a half.

Table 3. Means, standard deviations and t values for percentages and time on line for faculty and students taking the Webster's Diacritical Marking Computer Assisted Instruction Program.

Measurement	Faculty (n=40)		Student (n=55)		t
Percentage Gain	mean	42.15%	mean	31.92%	-3.574**
	sd	16.47	sd	11.44	
Time on Line (hours)	mean	1.41 hr.	mean	1.32 hr.	- .89
	sd	.43	sd	.51	

To evaluate possible relationships between gain and time on line and between gain and pre-test performance, correlations were statistically established as reported in Table 4.

Table 4. Percentage gain correlated with time on line required to complete the Webster's Diacritical Markings CAI Program and pre-test scores.

	Time on Line	Pre-test Score
Student Gain	.261	- .778**
Faculty Gain	.551**	- .920**

Faculty gain was found to be significantly correlated with faculty time on line (r .551**). Student gain versus time on line approached but did not reach the 5% level of significance (r .261). In general, then, greater gain tended to be associated with more time on line.

Student and faculty percentage gains were negatively correlated with pre-test scores (r -.778** and r -.920** respectively). The students and faculty members with lower entry scores tended to show proportionately greater gain.

Profile ratings for hearing discrimination and for speech intelligibility were also correlated with percentage gain in the student group. However, it became evident that the audio unit used with the program was erratic in performance. There was no significant correlation between gain and speech intelligibility in the student group.

Conclusion. The objective, to provide cognitive skills in recognizing and using selected diacritical markings, was successfully achieved by both the newly employed NTID faculty and for NTID students. One hundred percent of the new faculty and 88% of the students met or exceeded the minimal acceptable level of achievement (80% of all criteria). The CAI program effectively met the diverse individual needs of both students and faculty. No significant differences between the two groups were found relative to entrance scores or average time required to complete the course. Exit scores and percentage gain were both significantly higher in the faculty group.

This report has demonstrated individualized instruction through Computer Assisted Instruction is effective in providing the specified skill to both students and faculty.

ACKNOWLEDGEMENT

The authors acknowledge Ms. Marilyn Nutter anc Ms. Ella Klueck for their work in developing this program under the direction of Dr. James von Feldt.

Dr. Fred Wilson's assistance in statistical analysis of data is also acknowledged with gratitude.

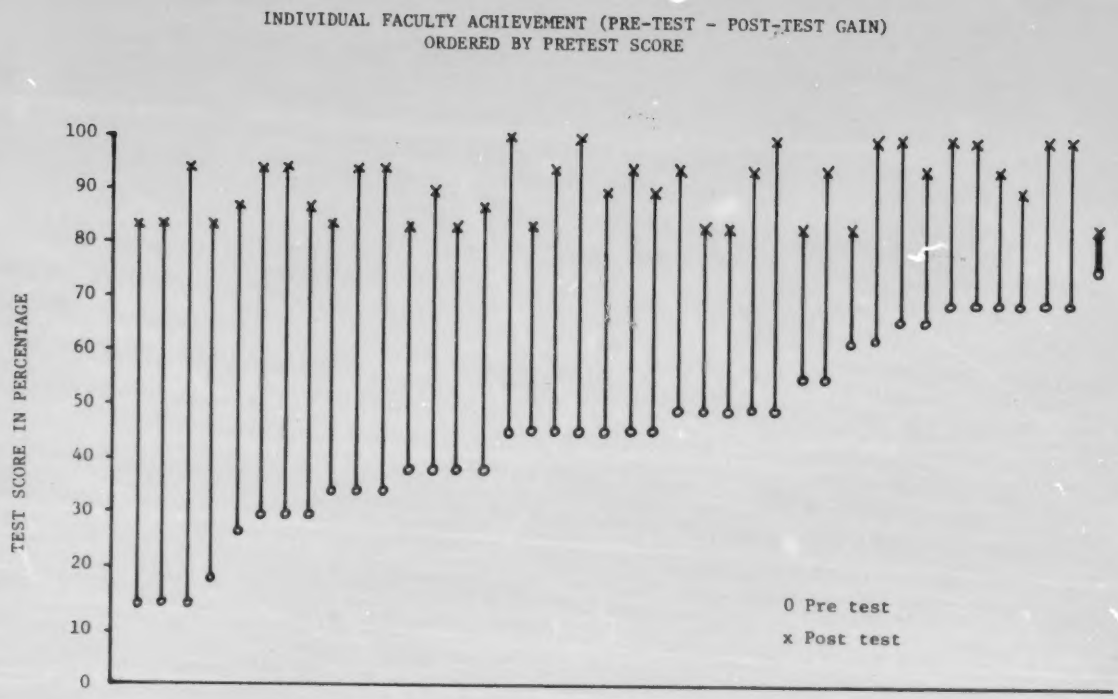


Figure 1. Pre- and Post-test Scores Graphed Individually for Faculty Group.

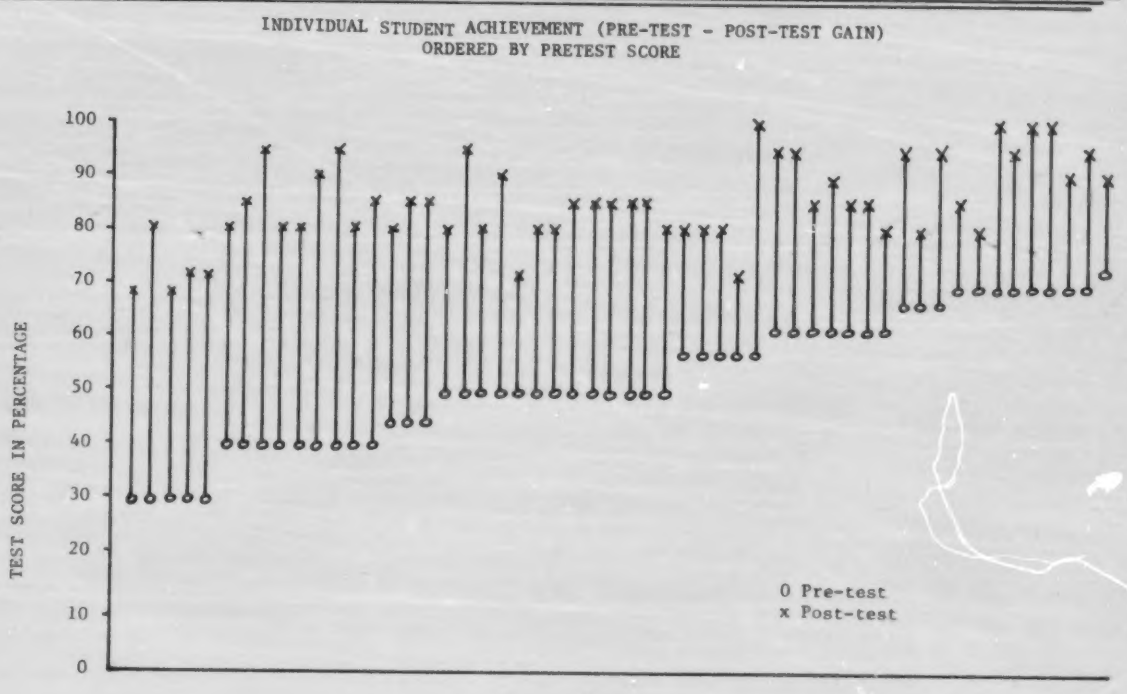


Figure 2. Pre- and Post-test Scores Graphed Individually for Student Group.

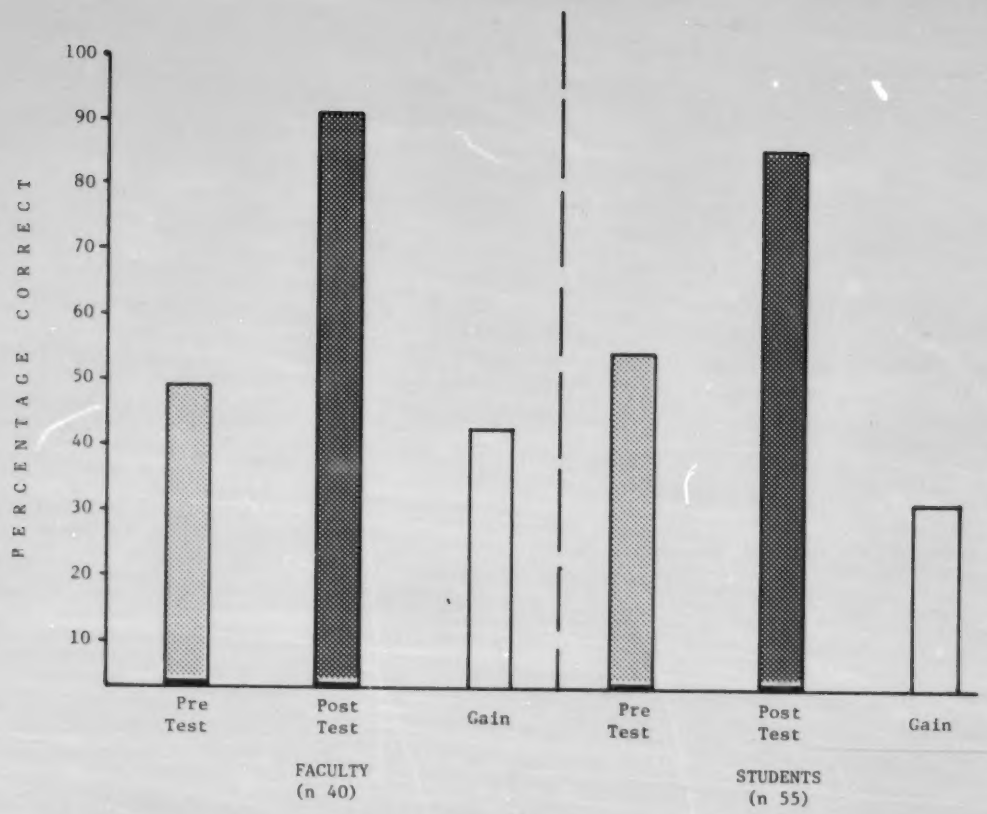


Figure 3. Mean test scores and percentage gain for faculty and student groups before and after completing Webster's Diacritical Marking Computer Assisted Instruction Program.

COMPUTER-BASED BUSINESS GAME IN SENIOR COLLEGIATE "CAPSTONE" COURSE

Frank L. Kaufman, D.B.A.
School of Business and Public Administration
California State University, Sacramento 95819

ABSTRACT: This paper addresses the problem: "Does a computer-based business game improve learning in the senior-level Policy course?". More than five years of experimentation with the use of a computer-based business game in the senior collegiate "capstone" course has yielded limited data which do not show clearly that a computer-based business game improves learning in the Policy course. For example, the class grade point averages obtained in the course with and without the game are identical. However, the data do show that the results obtained from the game (expressed as rankings) are consistent with results obtained by usual grading practices, so the game appears to be a valid addition to the course. Further, the game appears to generate added interest in the course, and for that reason may add to knowledge not measured in grading of performance in class. The last two points mentioned above encourage the author to proceed with further experimentation and expanded research in the form of a follow-up survey of graduates.

The Problem

Does a computer-based business game improve learning in the senior-level Policy course?

This question is one the author has sought to answer during eleven semesters of teaching the course called Business Policies and Management at California State University, Sacramento. In business schools literature this course is commonly referred to as the Policy course.

Background

The Policy course is the senior-level "capstone" course which is intended to teach students to integrate all the knowledge they have gained in their business and other courses up to the senior level by identifying, analyzing and solving complex business and government management problems. It is designed to provide the critical instruction and practice needed to help the students learn to "put it all together" in real-life situations.

The Policy course is an important part of the student's professional education for management. This is evident from the fact that the course is a key requirement in the curriculum fostered by the American Association of Colleges and Schools of Business (AACSB). The AACSB is the professional association of business schools which sets standards for professional education, and accredits those schools which meet the high standards. (Harvard and Stanford are examples of schools accredited by the AACSB). Our School of Business and Public Administration at CSUS is accredited by the AACSB in both the undergraduate and master's business programs, so one can see that presenting a high quality Policy course is important to us.

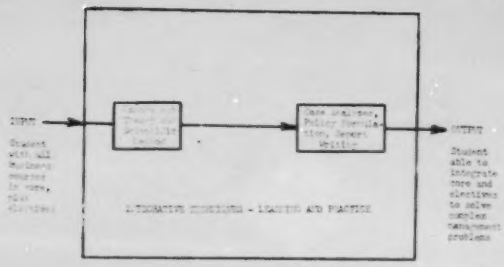
What is needed is a college course that simulates reality and demands the use of all of the tool skills that the student has learned. Only one semester is available for this in most curricula. The student must be challenged to bring to bear what he has learned in accounting, finance, behavioral science, business law, statistics, economics, communications, marketing, production and operations management, and all his general education in what amounts to a top management view of problems. His challenge must extend across an entire business or government organization, not just one function of it.

Alternative Ways of Teaching the Policy Course

The course is usually taught by the case method used at the Harvard Business School, and is modelled after the course taught there. Thus, the case method approach is the first logical alternative for teaching the Policy course. One should note that at Harvard it is offered only to Masters candidates in their second year of study, and it is a full year course.[1] The usual undergraduate course which is offered to seniors must, therefore, scale down the Harvard approach both as to time (one semester), and difficulty of cases.

At CSUS we teach principles of management and the scientific approach to problem solving as keys to the identification and solution of complex management problems. Figure 1 below is a visual illustration of the process.

The cases presented for student analysis are descriptions of actual management problems which have occurred in a wide selection of business and government organizations and have been written up



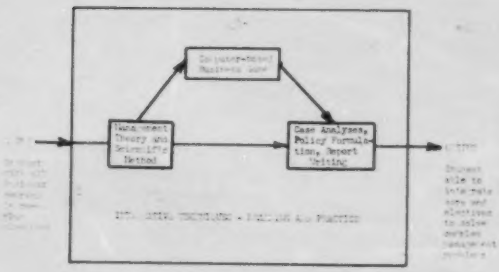
in case histories. They are "Canned" problems. The student is required to use all his knowledge to analyze the complex factors reported by the case writer in order to put his finger on the key problem and to devise an effective solution for it.

As mentioned above the scientific method is taught as the way to attack these problems systematically, i.e. 1) define the problem; 2) gather, analyze and evaluate the relevant data which illuminate the key problem and point to factors needing correction; 3) postulate and evaluate alternative solutions; 4) select the best of these; and 5) devise a plan of action to carry out the best solution. Naturally, to be of any use to management the above must be written up, in essentially the format described above, to provide a record and to lay out the plan for corrective actions.

The advantage of the case method is that is gives students practice not only in integrating knowledge previously gained for useful attacks upon actual management problems, but also in writing up their findings. We hear repeatedly from employers who hire the graduates of our universities that their weaknesses lie in identification of key problems in management situations and in writing, so it can be seen that the Policy course is an important effort to remedy those defects.

A disadvantage, detected by many colleagues and myself, is that students sometimes regard the "canned" cases as not fully realistic and perhaps distant from their own experience and goals. Many also feel that in a "real-world" situation they would be able to get much more data than the casewriter presents. We try to instill confidence in them by showing them that a lot of relevant, informative and comparative data may be found in business and government publications about similar organizations, that reasonable assumptions may substitute for actual data, that management literature provides many standards for comparison, and finally that no manager ever had all the information necessary to make a timely decision. Nonetheless, for many the situation lacks a certain reality and relevancy. Moreover, students never have a chance to test their recommendations for corrective action. Instead, the instructor renders a judgement on the feasibility of these.

Thinking about the disadvantage mentioned above suggests that some tactic is needed to introduce more realism into the course. So, a second alternative might well be to add to the usual "canned" cases a case in which the student takes part, and in which he may gather data, make decisions, see the results of those decisions, and analyze his own good and not-so-good actions. A computer-based business game offers such an opportunity. Figure 2 below is a visual presentation of the process.



A colleague had experimented with the use of The Executive Game[2] in the Policy course for several semesters. He used it as an adjunct to the "canned" cases to give some greater semblance of actual experience. He felt some uncertainty about its usefulness, but found some benefits from it, also. One should note that there is some opinion opposed to the use of games. T.R. Martin, Dean of the School of Business at Marquette University wrote of this in an article in the AACSB Bulletin in 1973.[3]

An advantage of this alternative should be that the students are injected into a situation where they must make operational choices in a competitive situation and the rules of the game, expressed in a computer program with resulting print-out, indicate how successful their strategies were in competition with classmates. In other words, students are not simply trying to come out well in relation to a standard set by the professor, but are testing their knowledge and analytical skills against those of their peers.

Thus, "practical" experience gained by managing a company in The Executive Game provides basis for post-game analysis of management effectiveness in a dynamic situation. The analysis, based upon the scientific method for problem solving, leads to effective report writing. The how and why of analyzing "canned" cases should become more clear, and students should gain more confidence and capability with this approach than with the conventional approach which relies solely upon "canned" cases.

A disadvantage of this alternative is that students are obliged to devote more effort to this exercise than to the analysis of "canned" cases alone. Some say that this is an unacceptable burden in their last semester in college, although most say that they benefited greatly from it.

Testing the Alternatives

Naturally, the first alternative above was the one which I adopted when I began teaching the Policy course. It worked well, and the students claimed that they benefited from it. But it was noticeable that some still believed that the experience was foreign to them and to their interests and goals. Further, I noticed, as my colleagues previously had, that undergraduate students show a certain reluctance to deal with figures. Graduate students have usually been out in the world of work for awhile, and show a realization that figures constitute the basis for the most significant measurements of their work. In fact, we have trouble making them consider anything but the figures. The undergraduates are quite the opposite, and seek to speak and write in generalities which do not subject them to the need for critical and specific defense of their positions.

I began to think of the second alternative (i.e. the use of a business game in tandem with "canned" cases). It occurred to me that using a computer-based business game could do two things: It could give students a sense of reality and involvement in a management situation in which they needed to make an analysis, and it would oblige them to deal with figures. It would have the additional advantage of providing a real-world environment in which their performance vis-a-vis their classmates would require them to set objectives, examine external factors, and plan, organize, and control their efforts as they would do, ideally, in business. This latter would enable them to see and feel at first hand the importance of these efforts in the successful operation of any organization.

Accordingly, in the third semester of teaching the Policy course, I introduced The Executive Game[4] into the course. I determined to use it as a real-work case in which the students would be involved. They would make all the decisions, gather and analyze all the data they felt they needed, and write up a report upon it as their final examination. The report was typical of those which all company chief executive officers must submit to their Boards of Directors at the end of each year, viz. an end-of-the-year report saying how the company fared in the past year and why, and what actions were planned for the coming year to do better.

The results were very gratifying. Students did become involved personally in the game. The competition in the game saw to that, and made it seem real instead of "canned". A rapport among the students developed which had not been there before. Moreover, most of them began to understand figures, ratio analysis, the need for forward planning and control, and the need to gain all the information they could about the competition and the "external environment". These observations and the greater enthusiasm of the students made it seem that this alternative was better than the first. Further, some students have returned to report great success on the job because of lessons learned in this course.

But casual observations and some evident student enthusiasm are not substitutes for objective data. Accordingly, I have begun to acquire objective data upon the use of this alternative. To date, it is very tentative and incomplete. Nevertheless, it may be of some interest to report upon it at this time. A small research grant has been applied for in an effort to gain more definitive data in the future.

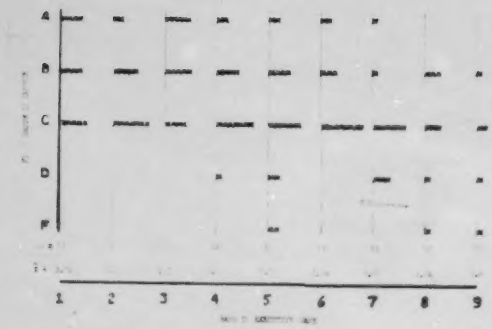
What has been done so far is this:

1. The Executive Game was dropped during the tenth semester of my teaching of the course in order to assess if it really did make a difference in class spirit and motivation. I believe that it did. The students in the two classes taught were not so enthusiastic and interested in their work as those in which the game was used. Some students, having heard from former students about the game, expressed disappointment at its ommission. Reinstated in the eleventh semester, the game resulted in greater motivation.

2. Each student's opinion about the usefulness of the game has been solicited as a part of the student's evaluation of the instructor. (At our School every instructor is evaluated by his students every semester as one input to the annual evaluation made by his colleagues on his teaching effectiveness and contribution to our School). The responses from the student survey have not yet been tabulated by our computer center, so there is nothing to report of this reaction at this writing. However, many students have stated at "rap" sessions held at the end of each semester that they gained a great deal from their experience with the game.

3. Information has been gathered for the last three semesters during which the game was used comparing the students' final grades with the ranking that they achieved in The Executive Game. Ranking in the game is determined by the rate of return on owners' economic equity. However, the final grade in the course is not influenced in any way by rank in the game. No points are given or taken away for high or low performance. Further, the final grade is made up of analyses and reports upon six to twelve cases (of which The Executive Game report is just one) plus grades for classroom attendance and participation. With that in mind, note in Figure 3 below that in the 110 instances covered here, there is a clear correlation between high rank and achievement in The Executive Game, and high final grades. This suggests that the game is a valid addition to the Policy course, tending to give results which are consistent with other achievement.

4. Final grades in all eleven semesters of teaching the Policy course have been compared to determine if there is any significant difference



between the class averages when the game was played and when the game was not played. Table 1 below shows the results of the comparison. Note that in classes where The Executive Game was not used, the average final grades were higher than those in which the game was used. However, I was aware (as teachers are often aware) that the motivation and aptitudes which the Fall 1975 group brought with them was significantly lower than during other semesters. They stayed low, relative to earlier classes, in spite of efforts to improve them, and the results the students obtained bear this out. Omitting the Fall 1975 group yields the interesting result that there is no difference whatever in grades when the game is used and when it is not.

TABLE 1

FINAL GRADES IN POLICY COURSE

Grade	Numbers of Students Earning Grades									
	1971 Fall*	1972 Spring*	1972 Fall	1973 Spring	1973 Fall	1974 Spring	1974 Fall	1975 Spring*	1975 Fall**
A	5	8	5	7	2	4	2	6	5	3	6
B	11	9	8	17	10	14	9	12	6	9	4
C	14	7	9	8	9	9	13	9	8	14	10
D	3	1	0	0	0	2	1	1	1	2	3
F	0	0	0	0	0	0	0	1	0	1	2
Total	33	25	22	32	21	29	25	30	20	31	25
Grade Average	3.03	3.20	3.05	3.06	3.43	2.93	3.28	2.87	2.75	2.35	2.19

* Indicates those semesters in which The Executive Game was not used.
** Indicates group below average in maturity and attitude.
*** Grade averages determined by assigning values to grades in usual fashion, i.e. A (4), B (3), C (2), D (1), F (0).

Grade average for all semesters in which the game was used = 2.62
Grade average for all semesters in which the game was used but excluding the below-average class of Fall 1975 = 2.71
Grade average for all semesters in which the game was not used = 2.71

Note that there is no difference whatever between grade averages when the game is used and when it is not,taking accounts of the exclusion of one below-average group.

Conclusion

The experimentation and observations reported upon above have not yielded a clear-cut answer to the problem stated at the outset, viz. "does a computer-based business game improve learning in the senior-level Policy course?" The general and specific observations which have been made yield results which are tentative, rather than conclusive. The data do support some tentative general conclusions, as follows:

1. The Executive Game, added to the conventional "canned" cases gives added interest to the Policy course by providing a more realistic problem than most "canned" cases provide.

2. The results obtained from the computer-based business game are consistent with the results obtained from the conventional approach to the Policy course.

3. There is no difference whatever between the final grades which the students achieve with or without the game.

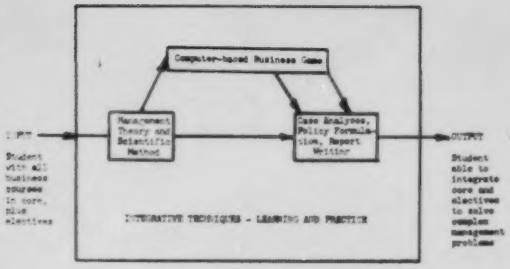
The limited data gathered so far do not clearly show that a computer-based business game improves learning in the Policy course. The data do suggest that the game heightens interest and involvement in the Policy course. Thus, it may be that the students gain broader and more direct (rather than simply vicarious) experience in planning for and controlling the operations of a nominal business, that is not gained with "canned" cases alone. That may increase knowledge, and enhance the usefulness of the course, but no direct data have been obtained on that question. Further research in the form of a follow-up survey of students who have completed the course could well be addressed to getting data to answer the question: "Did The Executive Game prove useful to you judging by later applications on the job of the lessons learned from the game?"

Judging from the data gathered thus far, it appears to be a "take it or leave it" proposition on the matter of using the game in the course. Leaving it would save a lot of time and effort for student and instructor. However, there remains the feeling that students do gain useful knowledge from the experience, and I feel further experimentation and research is warranted before abandoning the approach. Accordingly, I have asked for a research grant for a follow-up survey, and I have begun to experiment with the use of the game earlier in the course as a way to enhance appreciation of the principles of management by having the students begin at once to use those principles. That will be followed by the student's written analysis of the results of the game at about mid-semester, instead of waiting until final examination time. This may improve student insights into problems of "canned" cases, and improve his attitudes and aptitudes for writing up the "canned" cases. Figure 4 below is a visual presentation of that process.

There will be two reports on The Executive Game --one near mid-semester and one as the final examination. These will represent the first annual report of operations to the Board of Directors, and the second annual report. The plans for future action which appear in the first annual report will be carried into operation in the second year of the business game, and the effectiveness of those actions will be appraised and reported in the second annual report.

FIGURE 4

VISUAL PRESENTATION OF NEW APPROACH


Computer-based Business Game
Management Theory and Scientific Method
Case Analyses, Policy Formulation, Report Writing
INTEGRATIVE TECHNIQUES - LEARNING AND PRACTICE

Footnotes

[1]T.R. Martin, "The Undergraduate Capstone Course," AACSB Bulletin, Vol. 9, No. 2 (January 1973), p. 18.

[2]See Richard C. Henshaw, Jr. and James R. Jackson, The Executive Game, (Homewood, Ill., Richard D. Irwin, Inc., 1972).

[3]T.R. Martin, op. cit., p. 21.

[4]Henshaw and Jackson, op. cit.

COMPUTER SIMULATION FOR INSTRUCTION IN BIOLOGY

Benedict Mark Hall, Ph. D

Biology Department, Temple University, Philadelphia, Pennsylvania 19122

Abstract: To be useful as a teaching aid a simulation should not only be a valid model of its subject, but should be complex and unpredictable enough to be interesting. I will describe a simulation in plant development that must be manipulated by the user to produce a lifelike result, and two computer dialogues in genetics. The dialogues pertain to phenotype vs genotype in corn, and gene functions in E coli. The programs describe an experiment, ask questions, evaluate the answers, make helpful and relevant comments as needed.

INTRODUCTION: A computer can be programmed to carry out calculations, generate experimental data, carry on a dialog, ask questions, explain, comment on a student's answers, and keep score. Using the computer merely as a calculator would be wasteful of its capacity; the optimum use is to take advantage of its ability to follow branching programs, and to make responses based on the users input. Genetics and physiology are uniquely adapted to computer simulation because of the qualitative and quantitative logic of the subjects. The art of simulating for instruction depends on the degree of realism and variety that can be incorporated into the program. Realism depends on the depth of the analogy between the mathematical model and its subject. Variety is introduced by programming branches based on randomly generated numbers. Such control virtually guarantees that a student will never get the same question twice in one session at the teletypewriter. Even such trivial matter as being told whether his answer is right or wrong can be varied so that he will find that there is more than one way to say "you goofed."

My experiences with time sharing have involved a class of two hundred, divided into laboratory sections of approximately sixteen. Each laboratory is equipped with two teletypewriters. The students are encouraged to use the computer as an aid to calculations and also are assigned a number of computer dialogues as drills or quizzes. Each dialogue exists in the context of the laboratory exercises as they are explained by the laboratory manual in use, but the programs are provided with further explanations that the student may draw on or may bypass.

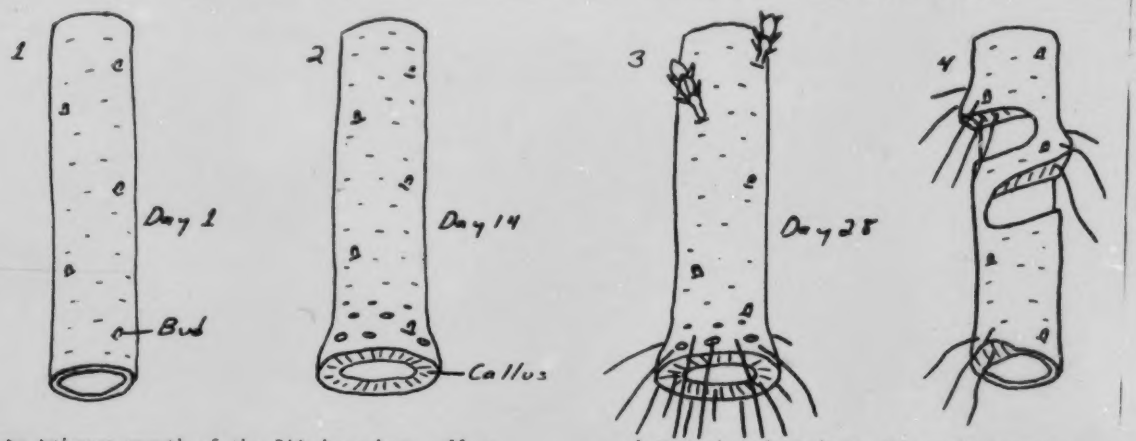
Student response runs the gamut from the immature responses of frustration, or delight with the new toy, to an intelligent appreciation and integration with the rest of the laboratory experience. To a considerable extent, the students are guided by the attitude of the laboratory instructor. The temptation to use the computer as a toy must be mildly discouraged: there is always one student in each section who has heard of "Startrek" or the football games that can be played with the computer. I have not attempted to teach the students programming, but any that show an interest are encouraged, and some student-written programs have been incorporated into the program library.

A feature of the computer that turns off some students is its impersonality and its intolerance for the errors of syntax or spelling that come easily to people unused to typing. I have attempted to make the computer seem more friendly and tolerant in several ways. On a trivial level the responces to both right and wrong answers have been made to sound informal and have an unexpected variety. On a more substantive level programs have been written to forgive small errors in spelling, or a student is given a chance to repeat an answer when it does not fit into the recognized spectrum of right or wrong answers. Finally, the dialogues have been made to help the student by commenting on mistakes in such a way as to guide his thinking in the right direction. The following is a description of three of the programs that I have been using in Botany and basic Genetics courses for the last four years. I will describe these programs in detail, and attempt to elucidate the biological principles involved as well as the programming strategy.

Time sharing has been with the Temple University Control Data Corporation Model 6400. We have previously used standard teletypewriters, and have just recently converted to Dekwriters. All the programs I have written for the courses have been written in BASIC. All the experiments simulated had been actually performed by the students in the courses; but it is probable that the simulations could stand alone, given the proper background reading or lectures.

RADGRAF: BIOLOGICAL PRINCIPLES. This program is a simulation of regeneration in a cutting of plant tissue. When a cutting- a segment of plant stem tissue- is excised from the parent plant, it will proceed under favorable conditions to regenerate a miniature, but complete new plant. This is accomplished through the development of a wound-healing 'callus' at the basal end and the release of buds from inhibition at the apical end. Eventually, when the callus differentiates roots, regeneration will be complete. Normally, these steps require only sufficient moisture to prevent desication, and auxin- the hormone that specifically stimulates the growth of the cells that form the callus. Auxin (indole acetic acid- IAA) is synthesized at the apex of shoots, and translocated downwards towards the base of the plant, being gradually depleted along the way. Consequently, cuttings of young wood may contain sufficient IAA

Figures 1-4: Effect of basal translocation of IAA applied at apex (see text)



to trigger growth of the IAA-dependent callus cells, but older wood may not. Roots differentiate from the relatively unorganized callus tissue, but only after its proliferation has slowed down or stopped. These stages of growth are under the direct control of IAA, as the concentration of the hormone that optimally stimulates the growth of callus and the development of root primordia, inhibits subsequent growth of roots. Root growth is permitted by an IAA concentration three orders of magnitude lower than that which stimulates callus growth (1,2). As a result of apical synthesis, downward transport, and gradual depletion, the concentration of auxin decreases from the apex to the roots. If an any point, the flow is interrupted by cutting or girdling of the stem, a local accumulation will ensue. (see Figures 1-4). The destruction of auxin is mediated by a system of IAA oxidases, which are present in all tissues, but particularly active in the roots. (2)

RADGRAF: THE PROGRAM. The simulation program assumes that auxin has been supplied to the cutting apically, dissolved in a paste of lanolin. The program quantitates the following parameters of the regenerating system according to values entered by the user:

1. The rate of translocation of IAA to the base.
2. The rate of IAA destruction in the stem, in the callus, and in the root tissue.
3. Values for optimum concentrations of IAA have been selected from the literature (1,2) for both callus and root tissue, and entered into the program.

Callus and root tissue grow when the auxin concentration is favorable, but roots do not start until sufficient callus (400 mg) has been formed and until the callus growth rate has been decreased to a low value.

The program recycles each "day" of the experiment, transfers a quantity of auxin from the site of application at the apex to the site of accumulation at the base of the stem segment. The amount transferred is determined by the user. Once translocated to the base, the auxin stimulates the growth of callus to a degree that depends on the ratio between actual and the optimum concentration. The destruction of auxin at these sites depends on the activity of IAA oxidases, also set by the user. Naturally, growth of a high oxidase tissue will increase the net rate of destruction, and hence tend to decrease the concentration at the stem base. The requirements for root growth, are prior development of at least 400 mg of callus, and a slowing down of callus growth. Manipulation of the program by the user is directed to the goal of matching the development of a normal cutting of Ligustrum or Syringia, which proceeds about as follows: (Table 1)

Table 1

Day	Event
1	Cutting made, IAA applied at apex
7	Callus growing
14	Callus growing at maximum rate
21	Callus growth completed, root growth starting
28	Roots growing rapidly
30	Program terminates

The program makes provision for two chief methods of output. After entering trial values of tissue parameters, the user may request a column printout of the results. This request will result in a printout of the "daily" values of six variables: apical and basal supplies of auxin, daily increments of callus and root tissue, and cumulative growth of callus and root. If adjustments are necessary, the user may enter new values of any parameter that he has the power to control, and request a new printout, or a scan. The scan allows the program to go quickly through a complete cycle without printing anything except a report on whether roots were induced to grow or not, and the total amount of roots, and the day on which root growth started. The user may request a plot of the results at any time; at this request the teletype will print out a graph of tissue growth. This is the most

Figure 5: Growth of cutting of plant tissue, simulated by the computer program RADGRAF. Plot shows basal translocation of Auxin, growth increments of callus and root tissue, and total callus and root accumulation. Horizontal axis, experiment "days" 1 to 35. Vertical axis, micrograms or milligrams, full scale (maximum values) as indicated in Legend (see text).

LEGEND

SYMBOL	PARAMETER	VARIABLE	UNITS	FL SCALE
$	A1	APICAL AUXIN	MICROGRAMS	5
*	A2	BASAL AUXIN	MICROGRAMS	1.96313
&	I1	CALLUS INCREMENT	MILLIGRAMS	58.894
$	I2	ROOT INCREMENT	MILLIGRAMS	53.939
+	G1	TOTAL CALLUS	MILLIGRAMS	678.54
@	G2	TOTAL ROOTS	MILLIGRAMS	176.057

aesthetically pleasing output. Scale of the graph is automatically computed such that the maximum value of each measurement recorded on the previous run is set to represent full-scale on the plot. When the user has discovered a set of tissue properties that produces a lifelike response of the tissue growth, then he has "won." There is, of course, no single winning combination, but a successful set seems to involve a rather low dose of auxin that translocates at a rate that builds basal supply to maximum within about a week, and is exhausted rather quickly as the auxin-destroying capacity of the growth increases. Proper adjustment of these early properties of the growth will allow the growth of roots to get well under way within the time limit. (See Figure5)

EZMAIZE, A DIALOGUE IN PHENOTYPE VS GENOTYPE IN CORN: BIOLOGICAL PRINCIPALS. The genetic control of phenotype in corn is a model for the complexity of developmental genetics in general. Genetic corn is excellent material for classroom demonstrations of Mendelian heredity. A satisfactory variety of ears are available (3), for demonstrations from the most elementary 3:1 ratio to much more subtle examples of epistatic interaction and linkage. The reference book on Maize by Neuffer, Jones, and Zuber (4) illustrates a number of mutants, and is good for laboratory reference for classroom exercies in corn genetics. The color of grains may be purple, red, yellow or white- controlled by 6 major sets of genes. Texture can be smooth, wrinkled, or shrunken- (4 sets). The storage carbohydrate of the seed may be sugar, or starch- (4 sets). The transparency of the seed may vary from translucent to waxy- (3 sets). The fact that the 17 effects mentioned above can be created by the action of

Table 2

Interaction of Genes Controlling Traits In Corn Seeds

Trait	Controlling Gene Loci
Color	A_1, A_2, C_1, R_1, Pr, Y_1
Texture	$Sugary_1$, $Sugary_2$, $Shrunken_1$, $Shrunken_2$
Transparency	$Sugary_1$, $Sugary_2$, Waxy
Stored carbohydrate	$Sugary_1$, $Sugary_2$, Waxy $Shrunken_2$

only 11 sets of genes is indicative of epistasis, the principle that genes interact to control traits. In Table 2, this interaction is classified according to the genetic locus involved.

The fundamental physiological explanation for the pattern of epistatic gene action is that a trait may require the sequential action of several gene-determined functions. In the case of the purple color of a grain of corn, for example, normal dominant alleles must be present at five loci: A_1, A_2, C_1, R_1, and Pr. A parallel situation is present in the example of storage carbohydrate: dominant alleles at 4 loci (of the eleven named) are required to be present in order that normal starch be formed in the growing seed.

Combinational analysis indicates 6×3^{10} different genotypic combinations between the 11 loci described, but due to gene dominance and interaction, the number of phenotypically different combinations would be only 72. The fact that transparency is difficult to recognize in the dark colored seeds further reduces the recognizable number to perhaps 36.

Figure 6. Sample printout of the computer dialogue EZMAIZE

```
GENOTYPE OF CORN GRAIN*
GENE SET   NO. OF DOMINANT GENES   NO. OF RECESSIVE GENES
A                  1                        1
C                  1                        1
R                  0                        2
I                  0                        2
PR                 0                        2
Y                  1                        1
S                  1                        1
SH                 2                        0
COLOR IS ? YELLOW
TEXTURE IS ? SMOOTH
RIGHT!
*

GENOTYPE OF CORN GRAIN*
GENE SET   NO. OF DOMINANT GENES   NO. OF RECESSIVE GENES
A                  2                        0
C                  2                        0
R                  1                        1
I                  0                        2
PR                 1                        1
Y                  0                        2
S                  0                        2
SH                 2                        0
COLOR IS ? PURPEL
TEXTURE IS ? WRINKLED
RIGHT!
*
```

EZMAIZE: THE PROGRAM. The program generates a variety of genotypic combinations, and asks the student to determine the phenotype. A random number function is used to generate the choices, and the selection is programmed in such a way that each of the different phenotypes appear in as nearly equal frequency as possible. After printing out a genotype the program asks the user for a statement of the phenotype, and comments on the answer. Since I have used this program to introduce the student to genetics and to computer dialogues, the program has been designed to allow for trivial errors in spelling. (See Figure 6)

CISTRON: BIOLOGICAL PRINCIPLES. CISTRON is a simulated genetic experiment with E coli in which strains mutant for two closely linked genes are mated. The program is modelled after an experiment in a laboratory manual written by Levine and Schwartz (5). The gene loci z and y are involved with lactose

metabolism: z^+ codes for an enzyme that splits lactose (a disaccharide) and y^+ codes for 'permease', a receptor molecule that facilitates the transport of lactose into the bacterial cell. Since both functions are required for the metabolism of lactose, neither the z- (deficient in b-galactocidase) nor y- (deficient in galactose permease) are able to grow on lactose as a source of carbon. The experiment makes use of the six varieties listed below (table 3.)

Table 3

Strain	Sex	Genotype	Streptomycin
1	M	$Fz\text{-}y^+/z\text{-}y^+$	Susceptible
2	M	$Fz^+y\text{-}/z^+y\text{-}$	Susceptible
3	F	z, y del.	Resistant
4	F	$z^+y\text{-}$	Resistant
5	F	$z\text{-}y^+$	Resistant
6	F	z^+y^+	Susceptible

In the experiment, matings are attempted by spreading a suspension of one of the strains on the surface of an agar plate and applying an overlay of the other strain on a small area of the agar. The agar medium contains the dyes eosin, and methylene blue, plus a low concentration of glucose, a higher concentration of lactose, and streptomycin. Due to the high concentration of lactose, any lac^+ cells grow rapidly, forming large colonies, which stain heavily. Lac- colonies grow more slowly and do not stain. The streptomycin prevents the male bacteria from growing, but allows them to donate chromosomes during conjugation. Complementary matings produce a merozygote (partial diploid) carrying both z^+ and y^+ factors; this cell will produce a dark blue (lac^+) colony. Non complementary matings such as $z\text{-}y^+$ X $Fz\text{-}y^+/z\text{-}y^+$ will produce rare lac^+ colonies because of recombination between different mutant loci withing the z- cistrons. Similar recombination may take place between mutant loci in the y cistron. Back mutation to the wild type from z- and y- will take place at even lower frequency.

CISTRON: THE PROGRAM. The program CISTRON outlines the biology and design of the experiment as described above, then proceeds to the description of a hypothetical mating as follows: "First, a plate of EMB agar is spread with the first strain of bacteria. The spread suspension is allowed to dry, then the plate is spotted with a drop of the second strain, and incubated for 48 hours. You will be asked to predict the kind of growth on the plate, and on the spot, and the density of the lac^+ colonies." The students are instructed to describe growth on the plates as "blank" (no growth), "-" (pale, lac-), or "+" (dark blue, lac^+). They are also instructed to report the concentration of lac^+ colonies as "none", "sparse", "numerous", or "solid" (confluent).

The program poses questions to the student by choosing randomly one of the 30 different strain combinations, then choosing one of the four possible questions. The answer given by the student is then compared with the seven possible responses and an appropriate comment is selected. In cases where the answer given is not recognizable, the student is told to "say it again". The right answers are identified, and a helpful suggestion is offered in the event of an incorrect answer. When the answer is wrong the student is given another chance before the program generates the next question. Since there are 30 mating combinations, 4 questions, and 7 answers, the total number of combinations is 840, of which 720 are inappropriate. Rather surprisingly, I found that a battery of 25 different comments were sufficient to say something relevant, true, and helpful as a response to each of these 720 different mistakes. Figure 7, below, is a segment of a dialogue with the program CISTRON.

Figure 7: Printout of the program CISTRON

FIRST, A PLATE OF EMB AGAR IS SPREAD WITH ONE STRAIN OF BACTERIA, THE SPREAD SUSPENSION IS ALLOWED TO DRY, THEN THE PLATE IS SPOTTED WITH THE OTHER STRAIN, AND THE PLATE IS INCUBATED FOR 48 HOURS. YOU WILL BE ASKED TO PREDICT THE KIND OF GROWTH ON THE PLATE, THE SPOT, AND THE DENSITY OF LAC^+ COLONIES.

*ASSUME THAT YOUR EMB PLATE WAS SPREAD WITH STRAIN 4 AND SPOTTED WITH STRAIN 1.

PREDICT THE GROWTH IN THE SPOT ? +

BULLSEYE!

INDICATE THE DENSITY OF LAC+ COLONIES AROUND THE SPOT ? NUMEROUS

NEVER!

MUTATION IS A VERY RARE EVENT.

INDICATE THE DENSITY OF LAC+ COLONIES AROUND THE SPOT ? SPARSE

RIGHT!

*ASSUME THAT YOUR EMB PLATE WAS SPREAD WITH STRAIN 3 AND SPOTTED WITH STRAIN 5.

INDICATE THE DENSITY OF LAC+ COLONIES AROUND THE SPOT ? SPARSE

IMPOSSIBLE!

MUTATION CAN'T RESTORE A DELETED GENE!

INDICATE THE DENSITY OF LAC+ COLONIES AROUND THE SPOT ? NONE

YOU ARE RIGHT!

DESCRIBE THE GROWTH IN THE SPOT ? -

SCORE ONE FOR YOU!

CONCLUSION: It may be early to assess the value of computers in instruction because the technology has not been in use for very long, compared to books, film, and television. However, the computer has the advantage that it is versatile, active instead of passive, increasing in power and speed, and decreasing in price. A computer program can be compactly stored and instantly retrieved. It is a permanent record but can be readily modified and replicated. It seems clear that a facility that can carry on an active dialogue with an individual student should have an important place in education and that its limits would

be determined more by the imagination of the programmer and the logic of the subject matter than by the technology of the computer.

REFERENCES

1. White, Philip R., 1963. The cultivation of Animal and Plant Cells, Second Ed. Ronald Press, New York.
2. Phillips, I. D. J., 1971. Introduction to the Biochemistry and Physiology of Plant Growth Hormones. McGraw-Hill, New York.
3. Carolina Biological Supply Co., 2700 York Road, Burlington, North Carolina, 27215.
4. Neuffer, M. G., Loring Jones, and Marcus S. Zuber, 1968. The Mutants of Maize. Published by the Crop Science Society of America, Madison, Wisconsin.
5. Levine, Louis and Norman M. Schwartz, 1973. Laboratory Exercises in Genetics, Second Edition. C. V. Mosby Company, Saint Louis.

A COMPUTER AND PROGRAMMING CURRICULUM FOR A SMALL COLLEGE

F. F. Liu

California State College, San Bernardino, CA 92407

ABSTRACT: We believe that every college graduate should be introduced to the ever increasing knowledge and the use of computer science. We feel that this is best done through programming courses where the student comes into immediate grips with problem solving. This college, a part of the California State University and Colleges system, is relatively new and small. It has no computer science related or engineering departments. This paper describes a Certificate Program in Computers and Programming presently made available to all students by the School of Natural Science. The paper traces the history of all computer related courses on this campus, the development of the present coherent program and its unique features including the use of microprocessors.

The purpose of this paper is to present a computer curriculum for a small liberal arts college within a much larger system. It is designed for students who need to have more competence than that of running a canned program in their own discipline. It is a program that can be completed by a student in a normal academic year along with his or her major studies.

The outline of the paper is as follows: a brief history of the college with special emphasis on subjects related to computer facilities and courses is then followed by a description of the program. A brief justification for the program is given and its special features are then described.

The California State College, San Bernardino, one of the 19 campuses of the California State University and College System, opened in 1965, offering six degree programs to just under 300 students. During its initial phase of development, the college focused upon a strong liberal arts curriculum. In the Fall quarter 1975, the college had an enrollment of 3988 students on the campus, a ten fold increase in a ten year period. However the projected enrollment increase for the next ten years is about 20%. It is one of the smallest campuses in the system with a total enrollment of just under 300,000 students and it is expected to remain essentially the same size for the foreseeable future. The realities of student preference in career objectives have dampened the zeal on liberal education and a number of career-oriented programs are now being implemented. The Certificate Program on Computer and Programming is one of these programs.

In the Fall of 1970, when I came to this college, it had an IBM 360/20 remote job entry station with an 8k byte memory. It was housed in a back room of a temporary student cafeteria building and was used exclusively for administrative job processing. It had neither the staff nor physical facilities for student or faculty use. In the following two years, valiant efforts were made by members of various departments, notably that of Mathematics, Physics, Psychology and Political Science to introduce computing to our existing curriculum without success. The failures were partly due to the inadequacy of the computer hardware and computer center personnel support, but I believe mainly due to the lack of a general introduction to the use of a computer to the campus community. It is hardly expected for a student in a class to learn about elements of programming on the side. There is just no time for this in a class whose course contents are already specified. For the vast majority of students who have the occasion to use a computer, an introductory course on a computer language is absolutely necessary, because there is just no time slot in his or her time schedule to learn about it otherwise.

In the Fall quarter of 1972, an introductory course in FORTRAN programming was given with the natural sciences students in mind by the Physics Department. When the student profile was studied, we find just as many students from other segments of the college enrolled in it. The course was therefore modified to accomodate other interests and made less mathematical in content. We limit the enrollment to 20 students to insure adequate access to keypunch and other facilities by the students. We introduce BASIC programming language in 1974, when a time-sharing facility became available. At that time we have available two teletype terminal ports linked by phone lines to the California State College at Northridge. These two ports are in contention with other 100 more ports from sister campuses across the state. This time-share facility was immediately so overloaded that its instructional function became greatly degraded.

In the same year, COBOL programming was offered for the first time. The student programs were run under batch and we have not been able to offer hands-on experience in this course, because of hardware limitation. Being one of the smallest campuses in the California State University and Colleges, we are clearly inadequate in terms of computing facilities. However, two recent upgrades bring great improvement on our campus. One in April, 1974, a Honeywell 2020 with a 40k byte word memory, a dual disk drive, a 200 line/minute printer and a card reader/punch was installed in the basement of a new library building. The other in February, 1976 when a local time-sharing system based on a PDP 11/45 with 8 available ports was installed.

A recent market survey conducted by the Division of Information Systems, California State University and Colleges revealed that of 1000 largest employers in this state 54% of the companies would require of a new graduate some computer related experience in his or her own discipline. In a related survey of the faculty of the 19 campuses of the California State University and Colleges,* it was found that a large shift towards the higher computer skill level is desirable in all disciplines. If this skill level is arbitrarily defined as: not applicable, indirect use, elementary programming, intermediate programming and advanced programming, the largest percentage of students should fall somewhere between elementary programming and intermediate programming by the year 1980. Of course, individual discipline will have varying desirable skill levels.

How does a small traditional liberal arts oriented college respond to this? With the projected no-growth enrollment up to the year 1985, the establishment of a new department is not very appealing to any administration especially in times of lean budget. Although our college has no engineering, computer science or data processing departments, it has on the faculty in various traditional departments people who have long experience in the use of the computer as a research tool. Therefore, we feel that we could initiate a program of practical instruction for our students. It is our premise that the majority of our graduates do not need the rigor and in-depth study of computer science, but should be exposed to courses in programming and computer use so that they will be a competent user of this powerful tool in their own discipline. With this in mind, we developed a Certificate Program in Computers and Programming by restructuring some of the old courses and introducing a few new ones. These courses are:

CSCI 100 COMPUTERS AND SOCIETY
CSCI 135 INTRODUCTION TO COMPUTERS AND BASIC
CSCI 140 FORTRAN PROGRAMMING
CSCI 340 ADVANCED PROGRAMMING
CSCI 345 COBOL PROGRAMMING
CSCI 350 COMPUTERS AND THEIR APPLICATIONS
CSCI 440 DIRECTED STUDY IN COMPUTERS/PROGRAMMING

AND ONE UPPER DIVISION APPLIED COURSE IN THE STUDENT'S MAJOR OR RELATED DISCIPLINE SELECTED IN CONSULTATION WITH AN ADVISOR.

A short description of these courses follow:

CSCI 100: COMPUTERS AND SOCIETY: 2.5 units. This is a survey course providing an introduction to computers, their capabilities, limitations, uses and misuses. Their revolutionary impact on modern society will also be examined. No previous background in computer or programming is required.

CSCI 135: INTRODUCTION TO COMPUTERS AND BASIC: 2.5 units. Introduces concepts of general usefulness to all programming languages with the use of the conversational-type language BASIC. Appreciation and experience in the use of a computer in a time-sharing mode is gained by programming in BASIC. No previous computer or programming experience is required.

CSCI 140: FORTRAN PROGRAMMING: 5 units. Introduces the basic elements of the computer programming language FORTRAN (FORMULA TRANSLATION). You will develop skill in analyzing, formulating, coding and debugging a series of programs related to everyday applications. No prerequisite, but CSCI 135 is desirable.

CSCI 340: ADVANCED PROGRAMMING: 5 units. Introduces programming techniques and data structure as exemplified in assembly language, I/O Programming, subroutines, co-routines, sorting and search techniques, stacks, queues, lists and symbol-state tables. Prerequisite: CSCI 140.

CSCI 345: COBOL PROGRAMMING: 5 units. Introduces the basic elements of the computer programming language COBOL (COMMON BUSINESS ORIENTED LANGUAGE). You will analyze, formulate, code, punch, and debug a series of programs related to the business world. Prerequisite: CSCI 135, 140 or ADMIN 220.

CSCI 350: COMPUTERS AND THEIR APPLICATIONS: 2.5 units. Introduces modern computers, a system view of their hard and softwares, reviews of large mini and micro computers together with their typical applications. Prerequisites: CSCI 135, 140 or 345.

CSCI 440: DIRECTED STUDY IN COMPUTERS/PROGRAMMING: 2.5 units. This course allows for individual projects or directed readings related to programming languages, mini and micro-computer systems and other related topics. The selection, implementation, documentation and presentation of the selected topics require prior approval of the faculty advisor in charge of the course. Prerequisites: CSCI 135, 140 or 345. It may be repeated for credit once.

A note of explanation about the units is in order. A class that meets four times a week for a total of 40 hours of lecture is given 5 units credit hours. Lab work required in most of these courses are not given separate credits.

The computer facilities available to the program include a CDC 3170 front ending a dual CDC 3300 computer system at the State University Data

*Interim Electronic Data Processing Master Plan 1975-1980. Division of Information Systems, Cal. State Univ. and Colleges.

Center accessible through the local Honeywell 2020 mentioned earlier. The local time-share system is dedicated to BASIC at the moment and may have assembler language available in the near future. In terms of hardware that the students can get their hands on, we have chosen a 16-bit microcomputer by National Semiconductors. Its language similarity to a minicomputer and its architectural simplicity as compared with other systems made it a very attractive laboratory equipment for student use. A complete system is small enough to be readily comprehensible and self-contained both in software and hardware and is priced under $8000. Our system consists of a teletype, a dual floppy disk, a PROM programmer unit, an 8k memory together with front panel display and control.

We plan to center our courses except those on high level languages on this microcomputer system and to introduce assembly language programming and basic knowledge on computer logic and hardware in this system. The learning of an assembly language with a restricted number of basic instructions under rudimentary or no operating system control can contribute much towards an understanding of a computer. We feel that it is at this level that our students in sciences will ever program in an assembly language. We also feel that basic knowledge on computer hardware and associated interfacing experience where the software and hardware meet can be most effectively illustrated in a microcomputer system at a minimum cost. We believe that the price of the microcomputer system will continue to fall and that several systems can be acquired within our budget limitations for students use in various courses and projects.

This Certificate Program was initiated Fall, 1975. It did not have enough time to prove itself yet. It has, however, met with strong student response, especially in the natural and social sciences. By the end of this academic year, all the courses in the program would have been offered at least once.

We conclude by iterating our belief that every college graduate whose discipline involves more than casual use of the computer can best be prepared by instructing them in high level programming languages together with directed individual studies with a small microcomputer system.

SIMULATION GAMING IN THE MANAGEMENT CURRICULUM - PRESENT AND FUTURE

William B. Hood
Dr. Richard M. Hodgetts

ABSTRACT: Simulation games are an integral part of the management curriculum. Since their introduction in the late 1950's their use and numbers have increased. Administrative and operational complexity have accompanied the increased reality offered by additional sophistication. Individual educators have innovated with games developed or modified for their particular needs. And all the while debate has continued over the costs versus advantages of gaming use. Results of a survey highlight the use and characteristics of management simulations and forecast their future in the management curriculum.

The application of simulation gaming techniques to the management education curricula had its foundings almost twenty years ago. (1,7) Its subsequent development and growth has been a function of societal changes, educational resources, individual exploration and discovery and the individualized management education programs conducted at the nation's colleges and universities. An absence of clearly defined leadership or even an acknowledged central clearing house for information has relegated this revolutionary teaching method to a random developmental pattern and inhibited the necessary acquisition and disbursement of available data concerning simulation's application to management education. Funds required to support in-depth study of simulation techniques and results have also been lacking due to the non-centralized nature of development that has been characteristic of management gaming.

Since its advent in the late 1950's, management simulation has stimulated an academic controversy between its opponents and supporters. Due to the lack of a recognized forum for debate on the subject, dissention and enthusiasm have been voiced in diverse publications, at sporadic intervals and on a variety of points of contention. The pros and cons on the subject have many times been concerned with incidental aspects of simulation gaming, but a major point of debate has been the overall pedagogical value of simulation techniques. While both sides have admitted to an increased enthusiasm and motivation resulting from game use, they have differed on the existence or degree of increased learning. (5) And, while opponents to management simulations have pointed out a lack of quantitative data supporting educational benefits, research has indicated that business games offer a new means of education that cannot be evaluated by traditional means. (2)

Further investigation and experience have disclosed that gaming's most valuable results are in integrating various management concepts, providing experience in decision making, forcing reliance on incomplete knowledge for decision formulation, clear illustration of the value of communication and organization, and as an educational tool of equal value to good and poor students alike. (6,7,11) Dissenters though, have pointed to the huge amounts of time and financial resources characteristically required for simulation activity and a continuing lack of quantitative verification of simulation attributes. Shortcomings in the areas of decision implementation, individual grading in a team project, lack of realistic organization and authority, and strategy formulation have been identified and highlighted by attempts on the part of some students to use the exercise only as an educational situation rather than a business experience. (3,10)

The latest claim made by simulation proponents involves the behavioral aspects of its participants. Concern has been voiced that the programs for management education have become quite well developed in management theory and quantitative methods at the expense of realistic problem solving and solution implementation. Acquisition of logical analysis and thought processes insofar as management activities are concerned has become a desirable goal for a modern management educational program. (8)

W. B. Hood is a Member of the Technical Staff, Hughes Aircraft Company, Buckley Air National Guard Base, Aurora, Colorado. R. M. Hodgetts is a Professor of Management at the University of Nebraska, Lincoln, Nebraska, serving as a Visiting Professor of Management, Texas Tech University, Lubbock, Texas.

OPERANT BEHAVIOR, AS THIS QUALITY OF MANAGEMENT ACTIVITY IS KNOWN, IS REPUTED BY GAMING ENTHUSIASTS TO BE THE MOST VALUABLE CONSEQUENCE OF A MANAGEMENT SIMULATION EXERCISE. WHETHER FOR LACK OF REBUTTAL TIME OR LACK OF DISAGREEMENT, OPPOSITION TO THIS CONTENTION HAS BEEN UNNOTICEABLE. LACK OF A VIGOROUS REBUTTAL ON THIS ASPECT OF MANAGEMENT GAMING IS INDICATIVE OF THE CURRENT CLIMATE OF GENERAL ACCEPTANCE AND USE OF SIMULATION GAMES IN THE MANAGEMENT CURRICULA AND ABSENCE OF INTENSE OR FORMIDABLE OBJECTIONS BY GAMING OPPONENTS.

THE HISTORY OF CONFLICT OVER SIMULATION'S PEDAGOGICAL VALUE HAS PRODUCED VALUABLE EXPERIMENTAL RESULTS APPLICABLE TO GAME DESIGNS AND THEIR IMPLEMENTATION INTO THE CURRICULA. EVALUATED DATA HAS INDICATED GAMING EXPERIENCE PROVIDES GREATER KNOWLEDGE AND APPLICATION SKILL OF MANAGEMENT CONCEPTS AND TECHNIQUES. INCREASED ENTHUSIASM AND MOTIVATION NOTICED IN SIMULATION PARTICIPANTS HAS SPAWNED MORE FAVORABLE ATTITUDES TOWARD MANAGEMENT THEORY AND PRACTICE AND HAVE BEEN COUPLED BY AN INCREASED LEARNING CAPACITY ON THE PART OF STUDENTS PREVIOUSLY DECLARED POOR LEARNERS OR DISCIPLINE PROBLEMS. GAMING EXPERIENCE HAS ALSO FOSTERED WITHIN ITS PARTICIPANTS AN AVERSION TO OTHER COURSES, INSTRUCTORS AND TEACHING METHODS, A FACTOR MORE EVIDENT AMONG STUDENTS INVOLVED IN THE COMPLEX GAMES WHICH DEMANDED MORE TIME AND EFFORT. (3,4,10) THESE EXPERIMENTAL RESULTS HAVE BEEN VERIFIED BY DIFFERENT SOURCES USING DIFFERENT DATA AND ANY SEEMINGLY CONTRADICTORY EVIDENCE HAS LACKED EITHER THE RIGOR OR ANALYSIS TO SERIOUSLY CHALLENGE THE MENTIONED CONCLUSIONS.

FROM PUBLICATION OF TWO GENERAL MANAGEMENT GAMES IN 1957 AND 1958, THE USE OF SIMULATION HAS DEVELOPED ON BOTH THE GRADUATE AND UNDERGRADUATE LEVEL OF COLLEGIATE EDUCATION. DESPITE ITS INITIAL CONTROVERSY AND INNOVATIVENESS, WITHIN THREE YEARS MANAGEMENT SIMULATIONS WERE IN EVIDENCE IN 78 PERCENT OF MANAGEMENT EDUCATION PROGRAMS WITH AN ADDITIONAL 13 PERCENT DESIROUS OF USING THEM. (9)

MEANINGFUL DEVELOPMENT FROM GAMING'S INITIAL STAGE TOOK TWO FORMS. MOST DRAMATIC SIMULATION CHANGES TOOK THE FORM OF COMPUTERIZATION. COMPUTER RESOURCES PERMITTED AN INCREASED GAME COMPLEXITY, NUMBER OF PARTICIPANTS AND VERSIMILITUDE DUE TO THE QUANTUM INCREASE IN COMPUTING POWER AND SCORING SPEED. THE DEVELOPMENT OF PRODUCTION AND FUNCTIONAL GAMES PROVIDED THE MOST IMPORTANT DEVELOPMENT, HOWEVER. BESIDES BEING MORE EASILY INTEGRATED INTO A WIDER NUMBER OF MANAGEMENT COURSES, THESE TYPES OF GAMES WERE LESS COSTLY IN TERMS OF TIME AND MONEY RESOURCES THAN THE GENERAL SIMULATIONS AND EXTENDED THE PEDAGOGICAL BENEFITS OF SIMULATION TECHNIQUES TO MORE AND DIVERSIFIED AREAS OF MANAGEMENT ACTIVITY. THIS GROWTH AND DEVELOPMENT HAS CULMINATED IN THE PRESENT STATUS OF SIMULATION GAMING IN THE MANAGEMENT CURRICULA.

TO MORE ACCURATELY ASSESS THE CURRENT STATUS OF MANAGEMENT GAME USE AND DEVELOPMENT AND TO PREDICT FUTURE TRENDS IN THE AREA, A QUESTIONNAIRE SURVEY WAS USED TO ELICIT PERTINENT DATA FROM POSSIBLE GAME USERS. (12) DATA SOUGHT INCLUDED INFORMATION ABOUT CURRENT SIMULATION ACTIVITIES, PLANS FOR FUTURE INVOLVEMENT WITH GAMING TECHNIQUES AND PERSONAL COMMENTS ON AND EVALUATION OF SIMULATION'S PRESENT AND PREDICTED PEDAGOGICAL STATUS. THE FOLLOWING ARE CONCLUSIONS REACHED IN THESE AREAS FROM THE QUESTIONNAIRE SURVEY.*

1. MANAGEMENT SIMULATION GAMES HAVE AN ESTABLISHED PLACE IN THE MANAGEMENT CURRICULA. THEY ARE ACCEPTED AND USED AS AN INTEGRAL MANAGEMENT EDUCATIONAL TOOL IN THE VAST MAJORITY OF MANAGEMENT PROGRAMS. THEIR VALUE IS REVEALED BY THE REQUIREMENT OF OVER 83 PERCENT OF THE GAME USING PROGRAMS THAT MAKE SIMULATION EXPERIENCE A PREREQUISITE FOR SUCCESSFUL PROGRAM COMPLETION.

HOWEVER, SIMULATION IS BASICALLY A COMPLEMENTARY EDUCATION TECHNIQUE RATHER THAN A PRIMARY PEDAGOGICAL APPROACH. THE PREPONDERENCE OF SCHOOLS HAVE TWO OR LESS GAMES IN THEIR MANAGEMENT CURRICULA AND EVEN THOSE THAT ARE USED ARE GENERALLY LIMITED TO LESS THAN HALF OF THE POSSIBLE CLASS TIME.

2. OF THE DIFFERENT TYPES OF MANAGEMENT GAMES, THE GENERAL MANAGEMENT SIMULATION IS THE MOST IMPORTANT TO CURRENT MANAGEMENT EDUCATIONAL PROGRAMS. ALMOST 80 PERCENT OF GAME USERS EMPLOY A GENERAL SIMULATION WHILE LESS THAN HALF HAVE PRODUCTION OR FUNCTIONAL GAMES IN THEIR CURRICULA. THE MULTI-VARIABLE PROBLEM, INTEGRATIVE ASPECTS AND ORGANIZATIONAL REQUIREMENTS OF GENERAL SIMULATIONS HAVE BEEN MUCH MORE IN DEMAND BY THE ACADEMIA THAN THE RELATIVELY LESS COMPREHENSIVE EFFECTS MODELLED BY PRODUCTION AND FUNCTIONAL GAMES. THE INCREASED TIME CONSUMPTION FOR BOTH STUDENTS AND ADMINISTRATORS AND INCREASED COSTS ASSOCIATED WITH GENERAL SIMULATIONS AS OPPOSED TO PRODUCTION GAMES, FUNCTIONAL GAMES, OR TRADITIONAL EDUCATIONAL TECHNIQUES HAVE BEEN OFFSET BY THE UNIQUE LEARNING CONCEPTS THAT ARE BELIEVED TO BE CHARACTERISTIC OF THE GENERAL MANAGEMENT GAME.

3. SIMULATION TECHNIQUES SEEM TO HAVE MORE VALUE AND APPLICATION TO UNDERGRADUATE ACADEMIC PROGRAMS THAN TO GRADUATE PROGRAMS. THOUGH USED IN SLIGHTLY MORE THAN HALF OF THE GRADUATE PROGRAMS, THE GAMES ARE USED IN ALMOST 80 PERCENT OF UNDERGRADUATE CURRICULA. THE ALLOCATION OF GAMING APPLICATIONS THAT IS IN EVIDENCE IS A PRODUCT OF SEVERAL FACTORS. THE GREATER TIME NECESSARY FOR AN UNDERGRADUATE DEGREE PROVIDES MORE OPPORTUNITIES FOR INTEGRATION OF SIMULATION GAMES WITH LESS OVERALL EFFECT ON PREVIOUSLY CONDUCTED ACTIVITIES. GRADUATE EDUCATION IS INVOLVED MORE DEEPLY IN ANALYSIS AND THEORY OF MANAGEMENT AND THUS IS LESS LIKELY TO SEEK THE INTEGRATIVE AND REALISTIC EXPERIENCES THAT WOULD BE EXPECTED ON THE "HOW DO YOU DO IT" LEVEL OF UNDERGRADUATE MANAGEMENT TRAINING. ANOTHER FACTOR INFLUENCING THE ACADEMIC LEVEL OF GAMES IS THE SMALL PROPORTION OF STUDENTS THAT GO ON TO GRADUATE STUDY. WITH THE MAJORITY OF COLLEGIATE GRADUATES COMPLETING ONLY THE UNDERGRADUATE LEVEL, IT IS REASONABLE TO CONCENTRATE SIMULATION IN THAT LEVEL SO AS TO PROVIDE AS MANY PEOPLE WITH THE

*TABULATED SURVEY RESPONSES ARE PRESENTED IN APPENDIX A.

EXPERIENCE AS POSSIBLE. THESE FACTORS EXPLAIN THE UNEQUAL USE OF MANAGEMENT GAMES BY THE DIFFERENT LEVELS OF EDUCATIONAL CURRICULA, BUT DO NOT MAKE A CLAIM FOR OR AGAINST THE CURRENT NUMERICAL ALIGNMENT.

4. THE EXECUTIVE GAME FROM UCLA IS BY FAR THE LEADER IN SIMULATION GAMES. ITS USE EXTENDS TO 25 PERCENT OF THE SCHOOLS USING SIMULATION GAMES. THIS MOST WIDELY USED GAME IS THE LEADER IN THE COMPLETE SIMULATION FIELD AS WELL AS THE GENERAL MANAGEMENT GAMES SEGMENT, IN WHICH IT IS CLASSIFIED. NO CLEAR OR PROMINENT HIERARCHY EXISTS AMONG THE OTHER GAMES IN USE, WHETHER THEY ARE PRODUCTION, FUNCTIONAL, OR GENERAL.

THERE IS NO DEFINED LEADERSHIP IN TERMS OF GAME AUTHORSHIP. ONE AUTHOR DID PUBLISH TWO GAMES BUT TOGETHER THEY WERE LESS COMPLEX THAN ONE GENERAL SIMULATION, SO HIS ACCOMPLISHMENTS DO NOT OVERSHADOW THE OTHERS. LEADERSHIP MIGHT BE BESTOWED ON THE AUTHOR OF THE UCLA EXECUTIVE GAME, BUT SINCE THAT IS HIS ONLY GAME IN GENERAL USE IT IS UNCLEAR IF ITS SUCCESS IS A RESULT OF ITS INNOVATIVENESS AND ACADEMIC DEVELOPMENT OR HIS CLARITY AND OPPORTUNISM.

SIMILARLY, THE PUBLISHING COMPANIES ALSO LACKED A LEADER IN THE FIELD. THE MAJOR GENERAL SIMULATION GAMES WERE THE ONLY GAMES BY THEIR RESPECTIVE PUBLISHERS IN USE, WHILE SEVERAL OTHER COMPANIES DID PUBLISH MORE THAN ONE GAME OF LESS COMPLEXITY OR USAGE. THE ABSENCE OF CLEAR SUPERIORITY IN GAME PUBLISHERS, AS IN GAME AUTHORS, IS INDICATIVE OF THE RELATIVELY BRIEF EXISTENCE OF SIMULATION GAMING AND THE LACK OF A CONCENTRATED, CONCERTED EFFORT TO DEVELOP ITS POTENTIAL TO THE MAXIMUM EXTENT POSSIBLE.

5. THERE IS A NOTICEABLE TREND TOWARD USING UNPUBLISHED MANAGEMENT GAMES AS AN EFFECTIVE WAY OF PROVIDING SIMULATION EXPERIENCE. OVER ONE QUARTER OF ALL GAMES CURRENTLY IN USE ARE GAMES DEVELOPED ON THE LOCAL LEVEL AND NOT PUBLISHED. LOCALLY DEVELOPED GAMES HAVE BEEN MORE NUMEROUS IN RECENT YEARS WITH OVER ONE-THIRD OF THOSE IN USE BEING DEVELOPED SINCE 1972.

CREATION OF GAMES NOT MEANT FOR PUBLICATION HAS BEEN ACCOMPLISHED PRIMARILY BY PROFESSORS AT THEIR RESPECTIVE INSTITUTIONS. IN THIS MANNER, GAMES HAVE BEEN DEVELOPED THAT HAVE UNIQUE ASSOCIATION WITH THE COURSES IN WHICH THEY ARE USED AND THE PARTICULAR CONCEPTS AND ACTIVITIES THE RESPONSIBLE EDUCATOR IS MOST INTENT ON IMPARTING. INCREASED FAVOR OF LOCAL GAMES HAS TO A LARGE DEGREE BEEN FOSTERED BY A GROWING KNOWLEDGE AND EXPERIENCE WITH SIMULATION TECHNIQUES BY MANAGEMENT EDUCATORS. IT HAS RESULTED IN A HOST OF GAMES WHOSE PARTICULAR ATTRIBUTES AND EDUCATIONAL BENEFITS ARE MORE CLOSELY IN CHARACTER WITH THE PARTICULAR INSTITUTION WITH WHICH IT IS ASSOCIATED.

6. MODERN MANAGEMENT SIMULATION GAMES CAN BE CHARACTERIZED BY THEIR COMPUTERIZATION AND DYNAMIC FORMAT. COMPUTERS HAVE INGRATIATED THEMSELVES TO MANAGEMENT EDUCATION BY THEIR INCREASED SPEED AND ABILITY TO HANDLE INCREASED AMOUNTS OF DATA. SUCH SOPHISTICIATION HAS PERMITTED A GREATER COMPLEXITY, AND ASSOCIATED REALISM AND BROADER APPLICATION OF SIMULATION TECHNIQUES.

COMPUTERIZATION HAS TO A LARGE EXTENT BEEN RESPONSIBLE FOR THE DYNAMIC FORMAT OF MODERN DAY MANAGEMENT GAMES. SOFTWARE PROGRAMS THAT FORM THE BASES OF GAMES ARE RELATIVELY EASILY DESIGNED TO ALLOW A FLEXIBLE RESPONSE BASED ON DIFFERENT AMOUNTS OR TYPES OF DATA. SUCH FLEXIBILITY IS ALSO POSSIBLE IN PROVIDING FOR RANDOM EFFECTS OR EFFECTS BASED ON DATA NOT AVAILABLE AT THE TIME OF SIMULATION DECISION MAKING. CURRENT FLEXIBILITY HAS ACHIEVED A POINT WHERE IT IS IMPROBABLE THAT STUDENTS CAN BEAT THE SYSTEM WITH ONE CONSISTENT STRATEGY AND THE CONSTANTLY CHANGING CIRCUMSTANCES MORE ACCURATELY REFLECT THE REALITY OF BUSINESS LIFE. THE DYNAMIC FORMAT IS A MAJOR REQUIREMENT TO ACHIEVE THE GREATEST BENEFIT FROM SIMULATION TECHNIQUES IN THE MANAGEMENT EDUCATION PROGRAM.

7. BENEFITS OF GAME USE HAVE INCLUDED INCREASED MOTIVATION; REALIZATION OF THE IMPORTANCE OF GOOD COMMUNICATION SKILLS; FORCING DECISION MAKING, OFTEN BASED ON INCOMPLETE INFORMATION; LEARNING TO ACCEPT PREVIOUS DECISIONS REGARDLESS OF THEIR QUALITY; AND A REALISTIC EXPERIENCE OF THE TEAMWORK AND RESPONSIBILITY NECESSARY FOR GOOD MANAGEMENT. THE MOST UNIQUE AND VALUABLE ADVANTAGES HOWEVER HAVE BEEN THE INTEGRATION OF INDIVIDUAL MANAGEMENT CONCEPTS WITH MANAGEMENT ACTIVITIES, THE ILLUSTRATION OF MANAGEMENT AS A MULTI-VARIABLE PROBLEM THAT DEFIES COMPLETE ANALYSIS AND DEMONSTRATING THAT ONE DECISION HAS AN EFFECT ON MANY AREAS OF A BUSINESS.

HOWEVER, MANAGEMENT SIMULATIONS USED IN THE VARIOUS EDUCATIONAL INSTITUTIONS HAVE DISTINCT DISADVANTAGES TO GO ALONG WITH THEIR ADVANTAGES. THE DISADVANTAGES THAT HAVE BEEN ACCEPTED IN ORDER TO USE SIMULATION TECHNIQUES ARE MORE IMPORTANT THAN THEY APPEAR AT FACE VALUE. LACK OF GAME FLEXIBILITY HAS SPURRED CREATION OF PERSONAL GAMES MORE APPROPRIATE FOR THE INDIVIDUAL CIRCUMSTANCE AND THE MATHEMATICS NECESSARY FOR ACCURATE SIMULATION CAN ONLY WITHSTAND ATTEMPTS TO BEAT THE SYSTEM FOR A LIMITED TIME. GREAT AMOUNTS OF TIME ARE TAKEN UP BY GAME PARTICIPATION. THIS INCLUDES TIME ACTUALLY SPENT DURING CLASS WHICH COULD BE OTHERWISE USED FOR DIFFERENT ACTIVITIES AS WELL AS ADDITIONAL OUT OF CLASS TIME NECESSARY FOR EDUCATORS AND ADMINISTRATORS TO ADEQUATELY PREPARE AND CONDUCT A CLASS INVOLVING SIMULATION. GRADING INDIVIDUALS INVOLVED IN A TEAM ORIENTED SIMULATION GAME HAS ALSO BEEN QUESTIONABLE AND ITS COMPLETE FAIRNESS AND OBJECTIVITY IS IMPROBABLE, THOUGH THE STUDENTS DO OBTAIN THE FULL BENEFIT OF THE GAME EXPERIENCE REGARDLESS OF THE GRADE THEY RECEIVE.

THE MOST VOLATILE DISADVANTAGEOUS FACTOR OF SIMULATION GAMES HAS BEEN THEIR ASSOCIATED COSTS. WHETHER IN DEVELOPMENT OF A SIMULATION, IN ITS ADMINISTRATION, OR IN ITS EXERCISE, COMPUTER SIMULATION GAMES ARE EXPENSIVE IN TERMS OF MATERIALS, TIME AND COMPUTER RESOURCES. THIS COST HAS BEEN HIGHLIGHTED DESPITE THE OVERALL HIGHER COSTS OF EDUCATION AND PROVIDES CONSTANT IMPETUS

TO STUDIES OF COST VERSUS RESULTS OF ASSOCIATED PROGRAMS.

8. THE COMPUTERIZED MANAGEMENT GAMES IN USE HAVE INCREASED IN SOPHISTICATION. THIS STATEMENT DOES NOT APPLY TO THE GRADUAL, OVERALL DEVELOPMENT OF SIMULATIONS, BUT RATHER TO THE INDIVIDUAL GAMES AS THEY ARE USED BY RESPECTIVE INSTITUTIONS. GAME ADMINISTRATORS HAVE IN A MAJORITY OF CASES MODIFIED THE GAMES THEY USE TO ACHIEVE A GREATER SOPHISTICATION WITHOUT AN OUTRIGHT CHANGE IN THE GAME USED OR ITS OBJECTIVES.

COUPLED WITH THIS SOPHISTICATION HAS BEEN A NOTICEABLE, AND SOMEWHAT CONTRADICTORY, CONCERN FOR GREATER SIMPLICITY AND FLEXIBILITY IN MANAGEMENT GAMES. CONTRADICTION IS EVIDENT IN THAT GAMES CANNOT BE MADE FLEXIBLE AND SIMPLE WHILE RETAINING A NECESSARY DEGREE OF VERSIMILITUDE; NOR DOES THE MAJORITY ACTION OF MAKING INDIVIDUAL GAMES MORE SOPHISTICATED SEEM CONSISTENT WITH A WISH FOR GREATER SIMPLICITY. THE OBSERVED FACTS OF SIMULATION USE HAVE ILLUSTRATED AN OVERALL GREATER SOPHISTICATION IN THE INDIVIDUAL GAMES IN USE.

9. SIMULATION EMPHASIS AND/OR USE SHOULD INCREASE IN THE NEAR FUTURE, BUT THIS INCREASE WILL NOT BE DRAMATIC OR EXCEPTIONALLY LARGE. OVER TWO-THIRDS OF THE MANAGEMENT EDUCATION PROGRAMS DO NOT PLAN ANY INDIVIDUAL CHANGE IN THEIR APPROACH TO SIMULATION. OF THE REMAINDER, THE LARGEST CHANGE WILL BE IN THE ADDITION OF NEW COURSES USING SIMULATION TECHNIQUES. ONLY A SMALL MINORITY OF PROGRAMS PLAN TO REQUIRE SIMULATION COURSE PARTICIPATION WHERE IT WAS PREVIOUSLY ELECTIVE, AND A NEGLIGIBLE PERCENTAGE ANTICIPATE DECREASING THEIR EMPHASIS.

CONTRASTING WITH THESE PROJECTED ACTIONS ON THE PART OF INDIVIDUAL MANAGEMENT PROGRAMS ARE THEIR PREDICTIONS ASSOCIATED WITH THE FUTURE OF MANAGEMENT SIMULATION GAMES. WHEREAS 67 PERCENT PLANNED TO DO NOTHING TO CHANGE THEIR UTILIZATION OF MANAGEMENT GAMES, ONLY 31 PERCENT PREDICTED THAT SIMULATION'S FUTURE USE IN THE MANAGEMENT CURRICULA WOULD REMAIN CONSTANT. ALTHOUGH A SMALL PERCENTAGE INDICATED MIXED FEELING ON FUTURE EMPHASIS AND USE OF GAMING, A CLEAR MAJORITY ANTICIPATED AN INCREASE IN THIS REGARD.

THE INDIVIDUAL PREDICTIONS REFLECT AN OBVIOUS PARTIALITY TO THE PERSONAL AND COLLECTIVE SIMULATION EXPERIENCE IF NOT AN ACCURATE RECORD OF DEFINITE PLANS OR DEVELOPMENTS. SHOULD A DYNAMIC AND MEANINGFUL LEADERSHIP EMERGE IN THE AREA OF MANAGEMENT SIMULATION GAMES, IT MIGHT BE POSSIBLE FOR THE FUTURE STATUS OF GAMING IN THE MANAGEMENT CURRICULA TO COINCIDE WITH THE BIASED INDIVIDUAL PREDICTIONS THAT ARE CURRENTLY EXPRESSED, HISTORICAL EVIDENCE, HOWEVER, INDICATES A CONTINUED SLOW, EVOLUTIONARY TYPE CHANGE IN GAME USAGE, CONTENT AND EMPHASIS.

REFERENCES

(1) ADLINGER, G. R. "BUSINESS GAMES--PLAY ONE," HARVARD BUSINESS REVIEW, XXXVI, NO. 2. MARCH-APRIL, 1958, PP. 115-125

(2) BOOCOCK, SARANE S., AND E. O. SHIELDS (EDS.). SIMULATION GAMES IN LEARNING. CALIFORNIA: SAGE PUBLICATIONS, 1968.

(3) BRASKAMP, LARRY, AND RICHARD M. HODGETTS, "PERSONALITY TRAITS AND SUCCESS IN A GENERAL MANAGEMENT SIMULATION," SIMULATION, XX, NO. 1, JANUARY, 1973, PP. 26-29.

(4) CALIFORNIA STATE UNIVERSITY AT SAN JOSE. "RESULTS OF GAME SURVEY" (A MIMEOGRAPHED SRUVEY).

(5) CHARTIER, MYRON. "AN EXPERIMENTAL STUDY OF THE EFFECT OF A SIMULATION GAME AND INSTRUMENTED DISCUSSION ON THE LEARNING OUTCOMES OF SUBJECTS." UNPUBLISHED DOCTOR'S DISSERTATION, UNIVERSITY OF DENVER, 1971.

(6) HODGET, RICHARD M., FRED LUTHANS, AND JAMES W. WALKER. "EVIDENCE ON THE VALIDITY OF MANAGEMENT EDUCATION," ACADEMY OF MANAGEMENT JOURNAL, XII, NO. 4, DECEMBER, 1969, PP. 451-458.

(7) _____. "MANAGEMENT GAMING FOR DIDACTIC PURPOSES," SIMULATION AND GAMES, I, NO. 1, MARCH 1970, PP. 55-66.

(8) KEYS, J. BERNARD (ED.). LABORATORIES FOR TRAINING AND DEVELOPMENT OF EXECUTIVES AND ADMINISTRATORS. OKLAHOMA: UNIVERSITY OF OKLAHOMA, 1972.

(9) KLASSON, C. R. "BUSINESS GAMING: A PROGRESS REPORT," ACADEMY OF MANAGEMENT JOURNAL, VII, NO. 3. SEPTEMBER 1964, PP. 175-188.

(10) LEWIN, ARIE Y., AND WESLEY L. WEHER. "MANAGEMENT GAME TEAMS IN EDUCATION AND ORGANIZATION RESEARCH: AN EXPERIMENT ON RISK TAKING," ACADEMY OF MANAGEMENT JOURNAL, XII, NO. 1, MARCH, 1969, PP. 49-58.

(11) RAIA, ANTHONY P. "A STUDY OF THE EDUCATIONAL VALUE OF MANAGEMENT GAMES," THE JOURNAL OF BUSINESS, XXXIX, NO. 3, JULY, 1966, PP. 339-352

(12) WOOD, WILLIAM B. "SIMULATION GAMING IN THE MANAGEMENT CURRICULUM." UNPUBLISHED MASTER'S THESIS, UNIVERSITY OF NEBRASKA, 1973.

APPENDIX A

Survey Data

Table 1

Types of Management Simulations in Use

Answers	Number	Percentage
General	50	59.5
Production	3	3.6
General & Production	19	22.6
Other	6	7.1
General & Other	3	3.6
All Type	3	3.6
Totals	84	100.0

Table 2

Levels of Management/Simulation Use

Answers	Number	Percentage
Graduate	17	20.24
Undergraduate	39	46.43
Both	28	33.33
Totals	84	100.0

Table 3

Status of Courses Using Simulations

Answers	Number	Percentage
Required	53	63.1
Elective	13	15.5
Both	17	20.2
No Answer	1	1.2
	84	100.0

Table 4

Class Time Involved with Simulation

Answers	Number	Percentage
100%	16	19.1
50% - 99%	8	9.5
20% - 49%	41	48.8
5% - 19%	19	22.6
Totals	84	100.0

Table 5

Unpublished Game Development

Answers	Number	Percentage
Pre 1968	10	41.6
1968 - 1971	6	25.0
1972 - 1973	8	33.4
Totals	24	100.0

Table 6

Origin of Unpublished Games

Answers	Number	Percentage
Professors	20	83.3
Graduate Students	2	8.3
Consultants	1	4.2
Programmer Employees	1	4.2
Totals	24	100.0

Table 7

Scoring Methods of Management Simulations

Answers	Number	Percentage
Computer Scored	85	91.4
Hand Scored	8	8.6
Totals	93	100.0

Table 8

Simulation Format

Answers	Number	Percentage
Fixed Format	12	14.1
Dynamic Format	73	85.9
Totals	85	100.0

Table 9

Class Response to Simulation Use

Answers	Number	Percentage
Increased Response	68	81.0
Mixed Response	6	7.1
Could Not Tell or No Response	10	11.9
Totals	84	100.0

TABLE 10

CHANGING EMPHASIS ON SIMULATION GAMING

ANSWERS	NUMBER	PERCENTAGE
INCREASE COURSES	21	25.3
REQUIRE ENROLLMENT	2	2.4
INCREASE COURSES & REQUIRE ENROLLMENT	3	3.6
NO CHANGE	56	67.5
OTHER - DECREASE	1	1.2
TOTALS	83	100.0

TABLE 11

INCREASED SOPHISTICATION IN GAMES

ANSWERS	NUMBER	PERCENTAGE
YES	48	57.2
NO	34	40.4
NO ANSWER	2	2.4
TOTALS	84	100.0

TABLE 12

HISTORICAL USE OF SIMULATION

ANSWERS	NUMBER	PERCENTAGE
STILL USING FIRST GAME	25	43.9
USED ONE GAME PREVIOUSLY	21	36.8
USED TWO GAMES PREVIOUSLY	5	8.8
USED "SEVERAL" GAMES	6	10.5
TOTALS	57	100.0

TABLE 13

PREDICTION OF FUTURE SIMULATION STATUS

ANSWERS	NUMBER	PERCENTAGE
INCREASE IN EMPHASIS/USE	48	64.0
STAY APPROXIMATELY THE SAME	23	30.6
DECREASE IN EMPHASIS/USE	2	2.7
MIXED OR INDEFINITE RESPONSE	2	2.7
TOTALS	75	100.0

TWO RECENT INNOVATIONS IN TEACHING:

A REPORT ON THEIR APPLICATION TO TEACHING RELIGION

William W. Wells
University of Hawaii at Hilo

ABSTRACT: A course in religion is necessarily pyramided; that is, introductory data must be acquired by the student early in the course in order for him to follow the discussion of the religious concepts which constitute the core of the course. The problem facing the instructor is to enable both the bright and the average student to acquire this data without boring the one or swamping the other. "Mastery Learning," a new idea in education, provides a possible solution to this problem, but demands that the instructor have access to many versions of the test over the introductory data. This latter hurdle can now be solved by using Computer Generated Examinations in tandem with Mastery Learning.

The introductory course in religion presents the instructor with the following dilemma: If he begins on the abstract, conceptual level where most of the really interesting religious issues appear, he can hope for only a limited understanding and appreciation of these issues on the part of the students because of their lack of a historical framework. How could one possibly help a student understand the Covenant in Exodus if he has never heard of a Hittite suzerainty treaty or the Code of Hammurabi? If, on the other hand, he begins with the factual framework which is a prerequisite to understanding and appreciating such issues, he runs the risk of losing the student to boredom before reaching the conceptual level. He may try to minimize this risk by devoting only enough time for the bright student to learn the appropriate background material, but then the average student will have difficulty in following the discussion when it moves to the conceptual level. Or, conversely, he may spend enough time for the average student to learn this material, but then the risk becomes a virtual guarantee of boredom for the bright student. In short, one group or the other is bound to lose when traditional educational methods are used. The ideal solution of this dilemma would be an educational program which would guarantee that all students--not just the bright--acquire the basic vocabulary appropriate to the topic of the course early in the semester without in the process boring the student who finishes the task quickly. By combining "Mastery Learning," a new concept from the field of education, with the technology of the computer, the instructor now has access to a solution which--if not ideal--is at least a step closer to that ideal.

All teachers are aware that aptitude for a subject is closely related to the amount of material that will be learned during a particular block of time. This perception leads one to expect to find only a few students in any class who will be able to master the subject and a few who seem unable to understand the subject at all, while most learn only a part of what is covered. But proponents of Mastery Learning insist that there is strong evidence to show that the aptitude of a particular student is also correlated to the amount of time that he will need in order to learn a given body of material to a given level. That is to say, if a student with less aptitude is given sufficient time, he is able to attain a level of achievement equal to that of the fast learner. "It is necessary to allow for the possibility that some students will never reach criterion.... But for most of the tasks in the regular school curriculum, it can be expected that every student will reach criterion if given enough time--a time within reason."[1] In other words, aptitude determines the rate of learning, not the amount which can be learned.

This thesis clearly has far reaching implications for the teaching of religion. A course of study in religion will almost always be sequential. For example, there is a body of historical and cultural background material which must be mastered before the student can read any particular portion of the Biblical text intelligently. Yet in a traditional undergraduate course in Old Testament the average student will study this background material, achieving only a partial grasp of it. He will take a test around the fourth week of class and receive a "C" grade, and then he will partially understand the lectures and

discussion for the rest of the semester. But if a "C" student, when motivated, can learn the historical background material as well as the "A" student learned it--albeit in a longer time frame, then it is grossly unfair to design a course in which he is precluded or discouraged from attaining that level of mastery. True, not all students will want to expend that additional effort. But that fact should not lead the instructor to design the course in such a way as to discourage this additional study. "Mastery Learning," then, is an approach to education by which all students are given the opportunity of attaining mastery of a body of material.

This modification in the [illegible] of the term "aptitude" demands a [illegible]onding change in the purpose of te[illegible]g. If all students are capable of mastering a body of material when given sufficient time, and if the instructor has clearly delineated the criterion of "Mastery," then testing becomes a means for finding out <u>when</u> that task has been accomplished. And since the student at the time of the test either has or has not reached the criterion established by the instructor, traditional grades become unnecessary for that block of study. The student either receives credit or he does not. And where credit is not earned, the test can be used diagnostically to guide continued study in the expectation that mastery will yet be achieved.

Such a change in the purpose of testing produces a change in the sociology of the classroom, in the relationships among the student, his peers, and the instructor. Since in this system each student is measured by a fixed standard rather than by his success vis a vis other students in the class, competition gives way to a concern for personal mastery of the material. The student is given a clearly defined task; it is no longer necessary, therefore, to fight with other members of the class for the coveted "A" grades. In addition, the instructor no longer appears as a Scrooge, jealously guarding a few "A's" and the "standards"; he can be seen as one willing to help the student attain a legitimately high level of competence.

An ideal subject for a Mastery Learning Program would have three characteristics: It would require minimal previous learning; it would be a subject which must be learned sequentially, and it would be a subject about which there is a good deal of agreement. While the third characteristic clearly would exclude much of what is discussed in a course in religion at the university level, nevertheless the introductory material in many university level courses in religion would satisfy these criteria.

With this as a background let me describe how I have embodied Mastery Learning techniques in teaching Old Testament Literature at the lower-division level.[2]

From the foregoing it should be clear that Mastery Learning lends itself most easily to a context in which one would normally use an objective type of examination. Thus I have limited my use of Mastery Learning to the first third of the course where the goal is clearly data acquisition. On the first day of class I distribute a study guide for the text and lectures which lists names, terms, and dates that I consider essential background for reading the Old Testament. Since there is no intrinsic virtue in surprise, and since I <u>do</u> know which specific names, terms, and dates I consider essential, the study guide is a complete list of that data. At the same time I explain briefly the following exam procedures:

1. The exam over the historical background will be objective and will cover <u>only</u> terms which appear on the study guide to the text and lectures.

2. It will be given on a Pass/No Credit basis; passing will be 85%.

3. A passing grade will be required for credit in the course; the student may take a second exam--and if necessary a third and a fourth--in order to achieve the stated criterion. This may be done during my posted office hours.

4. In all cases, the test must be passed by a specified date--usually about two weeks after the day on which I administer the test in class. (I added this last point after several semesters in order to emphasize the sequential nature of the semester's work. I had some students who tried to master the semester's material in reverse order!)

These procedures have considerably lowered the anxiety level of the students in my class and significantly raised the level of achievement in this particular area. I have been able to communicate the fact that there is a hierarchy of ideas in the course which must be recognized by both the teacher and the student; yet I have been able to keep the classroom emphasis on the ideas expressed within the Biblical text. And I believe that I have shown that I am on the side of learning.

Few instructors make it a practice

to allow students to retake an exam in order to raise a low score. But why should a student not be allowed this privilege? The refusal to allow another attempt favors those with high aptitude, that is, those who learn quickly. But the business of teaching deals with the acquisition of learning, not the bestowal of rewards on those with high aptitude. So, what exactly is the drawback to the practice that was outlined above? The only apparent answer to my question is the plea that test writing is time consuming. And that criticism is significant. I find that I need many versions of each exam that I administer under the conditions outlined above. With this criticism we pass on to the second major topic.

The problem of creating multiple examinations on a given topic can now be solved in any location having access to even a small computer through the use of Computer Generated Examinations. Let us suppose, for example, that I would like to give an examination with thirty multiple-choice questions which tests the students' mastery of ten specific subject areas. My first task would be to prepare nine or ten questions on each of the ten subject areas.

At this point the reader has probably noted the only major problem for the instructor in using Computer Generated examinations. The instructor must write 90 or 100 questions for a test instead of the thirty questions which would be required for a traditional examination. This requires a substantial amount of preparation by the instructor prior to using the test for the first time. In subsequent semesters, however, the instructor will need only to update and improve these questions. The questions are then keypunched and fed into the memory of the computer.[3] In order to begin constructing a test, the computer selects three questions at random from among the nine or ten available in the first subject area. These questions are stored temporarily on a designated memory area, and the corresponding answers to these questions are stored in a separate designated memory area. Then the computer picks three questions from the bank of questions on the second subject area and stores these also, and so on. When the process has been completed, the computer will have selected and stored thirty randomly selected questions, three questions from each of the ten subject areas. At this point the computer begins to print the first test. It begins by printing the title of the test and instructor's name at the top of the page along with the designation "Version 1." Then the questions are printed in a typical examination format. The computer can, for example, be instructed to print a blank space to the left of each question so that the computer printout itself can be used as the test copy. After printing the complete test, the printer moves to a new page, again prints the heading, including the designation "Version 1," and finally prints the answers which correspond to the particular questions which were used in "Version 1." The computer then clears the temporary question storage area and the temporary answer storage area, and begins the process a second time. It again selects thirty questions at random form the question/answer bank, and it prints a test and answer sheet designated "Version 2." Note that "Version 2" is a completely new test created from the bank of questions in exactly the same way as was "Version 1." Statistically the chances of "Version 2" being identical to "Version 1" are infinitesimally small. The process is continued until the computer has printed enough exams for the class. Thus each student in the class will have a unique test--for which the instructor will have a printed answer sheet. When ordering copies of a particular test, I ask for about twice as many copies as I need for the class in order to allow those students who do not achieve 85% on their first try to retake the test. After a particular examination has been used, it can be stored on tape or a disk in the computer center until the next semester.

In the preceding description I referred to multiple-choice questions; it will be obvious to the reader that True/False questions could be used as well. It should be pointed out that by making appropriate modifications in the program, a computer can also be used to create matching questions for a test. In this case, the computer chooses items for inclusion in a particular matching question from an item bank, randomizes the answers (and even throws in an extra answer or two from the bank of question/answer items), and then prints the questions and answers in a typical parallel column format. Questions on geography can be included. In this case the computer prints a random selection of place names from an item bank written by the instructor. (Answers will be numbers assigned by the instructor.) Then a map on which numbers have been placed in appropriate locations is reproduced and distributed by the instructor along with the computer printed test. The student is asked to write a number from the map in the blank which is provided on the computer printed test.[4]

The Computer Generated Examination without Mastery Learning would be merely a gimmick, except possibly for its utility as a control for cheating in large sections of a particular class. (Since each

student has his own unique examination, is virtually impossible to cheat on one of these examinations.) Mastery Learning without the use of the Computer Generated Examination could be a real burden on the teacher. But taken together, these two techniques provide a real alternative to the traditional approach when dealing with the specific problem of data acquisition. I have access to an unlimited supply of tests through the use of a computer. If the student is willing to spend several hours in additional study in order to retake an exam, then I am willing to invest a minute or two in test correcting.

Student reaction has been uniformly good. Initially there is some amusement at facing the "ultimate weapon" in the perennial battle between the student and teacher--a Computer Generated Examination. But students appear to appreciate the dual assurance that they will be able to attain mastery of the material and that the instructor will take the time to help actualize that goal.

1. John B. Carroll, "Problems of Measurement Related to the Concept of Learning for Mastery," reprinted in Mastery Learning: Theory and Practice, ed. by James H. Block (New York: Holt, Rinehart and Winston, Inc., 1971).

2. Mastery Learning by Block is an excellent introduction to the subject which includes extensive bibliographical material for the interested reader.

3. Since the details of programing would not be germaine to this discussion, I will limit myself here to a verbal description of the process from the instructor's point of view.

4. Gerald Lippey, ed., Computer-Assisted Test Construction (Englewood Cliffs, N.J.: Educational Technology Publications, 1974) is a good introductory work on Computer Generated Examinations. It includes extensive bibliographic data on recent publications in the field.

THE USE OF A HAND-HELD PROGRAMMABLE CALCULATOR AS AN AID TO CALCULUS INSTRUCTION

Martin T. Lang
California Polytechnic State University
San Luis Obispo, California 93407

ABSTRACT: The use of computers to facilitate the teaching of concepts in introductory calculus classes has received considerable attention in the last decade. Now with the availability of programmable calculators there are new opportunities to augment instruction in these classes, particularly to illustrate such concepts as functions, limits, derivatives, integrals and iterative techniques. This paper describes general procedures for illustrating these mathematical ideas with the help of a hand-held programmable calculator. Several specific examples are given in conjunction with the descriptions. Calculator requirements for such classroom use are presented and suggestions are given on the design and use of such programs.

In the last decade we have seen numerous attempts by mathematics educators to use computers to enhance our instructional program. At almost every level, from elementary school through college, experimentors have devised and tested programs where the mathematics instruction was supported or augmented by high speed computers.

Through these years computers have not only proliferated in our schools, but their configurations have changed dramatically. Now there is almost no reasonable basis of comparison in size, cost and operating facility between the present generation computers and their predecessors. Along with the cost reductions which have greatly benefitted our schools, probably the most noteworthy change has been the dramatic reduction in the size of computers. This miniaturization reached a sort of climax with the advent of the hand-held programmable calculators.

There are of course some people who would perhaps dispute the contention that these programmable calculators which are now being marketed in increasing numbers are indeed computers. The fact remains, however, that the programmability of these calculators, along with their ability to store and retrieve a significant number of constants, must qualify them as miniature computers, or at least as micro-computers. And as the miniaturization in electronic circuitry continues, these hand-held programmable calculators will no doubt increase in sophistication and will develop almost all the capabilities of their leviathan ancestors, including extensive storage memories and perhaps even programming languages. Recently IBM scientists developed some new FET memory chips which will make it possible to store up to five million bits of data on one square inch! It is exciting to imagine what this can do for the programmable calculators of the future.

Recognizing the tremendous potential of the programmable calculators, I would like to present some ideas concerning their use as an aid to calculus instruction. The remarks and examples given in this paper apply directly to our present generation hand-held programmable calculators. I will leave it to the reader to envision the extention of these ideas to the micro-computers of the future.

Computers in Mathematics Education

The use of computers in mathematics education can usually be categorized under two general headings: Computer Assisted Instruction (CAI) and Computer Extended Instruction (CEI). Without going into great detail describing these modes and at the risk of oversimplifying, let me simply say that in CAI the computer takes the role of the instructor so that it presents a particular learning sequence to the student much like a programmed text, while in CEI the computer becomes a tool for the classroom teacher to use as an augmentation of his instruction in an effort to clarify ideas and concepts. It is not my purpose to discuss the relative merits of CAI versus CEI in the Mathematics programs of our schools, nor do I intend to advocate one type of computer application over another, for it is clear that both modes of computer use have their place in our schools. However, my present concern is with CEI, and I would like to focus on the teaching of fundamental mathematical concepts in introductory calculus.

The idea of using the computer to enhance the teaching of concepts in calculus is of course not new. In recent years, a number of experimentors have developed programs of instruction in calculus which provided for computer augmentation to elucidate the fundamental ideas. Even formal educational studies were carried out by several researchers such as Fiedler (1969), Bitter (1970),

Holoien (1970), Bell (1972) and myself (Lang, 1973) which attempted to establish the validity of such an approach in calculus instruction. These studies, though not always conclusive, generally pointed toward an affirmative answer to the question: Is it effective to use computer extended instruction in introductory calculus? It seemed also transparent in the results of these investigation that the teaching of concepts such as functions, limits, etc., rather than techniques would be most positively benefitted with the computer augmentation.

With the availability of programmable calculators, it has become apparent that many of the things which were being done on large-scale computers to enhance calculus instruction could also be done with these "microcomputers". Furthermore, due to the facility with which a teacher can bring a hand-held programmable calculator into class, it seems likely that such an instrument can be most easily and effectively incorporated into the instructional program. As with any new mode of instruction some planning and experimentation will however be necessary in order to achieve any degree of success.

Calculator Requirements

But what capabilities are required of a minicalculator in order that it be an effective instructional tool in the calculus classroom? This is an important question which should be carefully answered before funds are expended for this purpose.

The first and foremost requirement is of course programmability. It must be readily possible to define mathematical functions, preferably two or three functions simultaneously. Functions of several variables should definitely be included in this capability.

Programming the calculator should be simple and natural. One essential feature is the ability to call subprograms. In most advanced programs, such as computing integrals, it is necessary to define the relevant function in a subprogram. Thus a particular program can be general in nature and may refer to user-defined function subroutines when necessary. Related to this is the possibility to loop in a program and to exit a loop when certain conditions are met. Calling subprograms or looping are particularly useful in performing iterations and repeated calculations.

The calculator should also be capable of a pause within a program. This is important since for instructional purposes it is often desirable to display intermediate results. Branching options are also essential and are used in almost every complicated instructional program. Branching decisions are often based on counters, magnitudes of numbers and algebraic size comparisons. Another useful feature is the ability to set one or more flags which then serve as a basis for branching. It is preferable that it be possible to set these flags both externally and through program control.

A memory for storing several constants is of course indispensible. At least five storage registers should be available and preferably considerably more. These are important for retaining intermediate results as well as for enabling the development of programs with multiple output.

Finally, the ability to record and save programs is a feature that is most desirable if not essential. Many desk top programmable calculators now provide this capability, but only a few hand-held calculators have this feature. Although it is possible to key into a machine even complicated programs using an instruction sheet, this can be very time consuming and frustrating. Programs stored on magnetic cards or tape are much more accessible and provide the flexibility needed for use in classroom instruction.

Design and Use of Programs

Because of the constraints on input and output on a hand-held calculator, it is extremely important that programs be carefully designed to facilitate their use. Several factors should be emphasized in this regard.

1. Programs should teach concepts. Therefore, intermediate results are often as important as the final result, and these should be readily displayed when so desired. A teacher may want to write these on the chalkboard as he steps through the various parts of a program.

2. Input should be natural, obvious and flexible. The user will find it easier to run the programs when changes in the parameters are easily made and when only minor modifications in a second run do not require unnecessary reinitializations of unchanged constants.

3. Output must be useful, clear and easily identifiable. In some calculators, the display mode can be changed under program control and this can be a helpful technique to distinguish various parts of an output list. For instance, if x_i, y_i are successively computed and displayed, the x_i's might be given with two decimal places while the y_i's are displayed with three decimal places.

4. Important intermediate results should be retrievable in memory registers. This might include a count of the number of iterations performed, or a partial sum or a result obtained on the previous "run" of the program.

5. Documentation of programs for your personal library is very important. Explicit descriptions of the procedures to use a particular program are necessary. The various types of output and the optional modes of running a program must be carefully explained. Specific examples as part of the documentation are always instructive.

6. Procedures for defining functions used in conjunction with a library program should generally be planned prior to a classroom demonstration. It is advisable to determine the exact key stroke sequence in advance to avoid embarrassing errors. In some cases, it may be easiest to

prerecord the function or functions needed with a particular program that illustrates a mathematical concept.

7. Programs should always be thoroughly tested before attempting to use them in the classroom. This will assure a smooth computer augmentation of the instruction.

Topics from Introductory Calculus Which Can Be Enhanced Using a Programmable Calculator

There are many topics in calculus where a programmable calculator can be a useful instructional tool. Below are presented some of the areas where it has been found to be useful to augment the instruction with a programmable calculator. It should be remembered that the primary emphasis is to use the calculator to teach basic concepts, and that its use as a computational device is secondary.

The ensuing remarks on how a programmable calculator can be used in the teaching of calculus are given in fairly general terms. However, in each case, working programs which accomplish the suggested presentation have been developed by the author for the HP-65 Programmable Calculator. Specific examples of these programs are provided in conjunction with each topic. The reader should find it straightforward to make the modifications necessary to implement these ideas on other sophisticated programmable calculators.

1. The Study of Functions

The concept of a mathematical function lies at the foundation of the study of calculus and it is of utmost importance that students develop a thorough understanding of it. The programmable calculator provides one of the easiest means of illustration functions. In fact, the calculator itself becomes the function-machine which associates with each element in the domain a particular element in the range of the function. Once a function-definition, say $y = f(x)$, has been programmed into the calculator, all that is required is to enter any number x from the domain of the function, to push a "RUN" button, and the resulting number will be the value of the function, $f(x)$. If a number is fed into the "function machine" which does not belong to the domain, the sophisticated calculator will give an indication such as a flashing signal or an "ERROR" message. If several functions are simultaneously definable on the calculator, then composite functions are easily illustrated by using the output of one function as the input of a second or third function. Inverse functions are illustrated similarly.

Example 1.
Assume that the functions A, B and C are defined by:

$$A(x) = x^2 + 2x + 1$$

$$B(x) = \frac{2x^2 - 2x - 40}{3x^2 + 15x + 12}$$

$$C(x) = \sqrt{25 - x^2}$$

If these functions are programmed simultaneously on the calculator under the function keys [A], [B] and [C] respectively, then we can readily observe the following:

$A(2.5) = 12.25$, $B(-3) = 2.67$, $C(4.2) = 2.71$, $B(-4)$ is undefined, $C(B(3)) = C(-0.33) = 4.99$, and $C(A(2))$ undefined.

(A flashing display or the message "Error" indicates the function is undefined. Note also that evaluation of the composite function $C(B(x))$ at $x = 3$ is accomplished with an input of 3 and pushing function key [B] followed by [C] to obtain 4.99.)

2. Graphing Functions

The programmable calculator can be a great facilitator in the graphing of functions, and it makes little difference whether we are using rectangular or polar coordinates. Once a function f has been defined, (i.e., programmed) on the calculator, one merely needs to input a set of values x_i from the domain and let the calculator compute $f(x_i)$. One can then easily plot the points $(x_i, f(x_i))$, to construct a graph. In some cases, especially when drawing polar graphs, it is advisable to input a set of values at regular intervals, say from $\theta = \alpha$ to $\theta = \beta$. For example, to graph the polar curve $r = f(\theta)$, we might choose angles θ_i from the domain in such a way that $\theta_{i+1} = \theta_i + \Delta\theta$, with $\theta_o = \alpha$, and $\Delta\theta$ some arbitrary constant. For this purpose it is convenient to program the calculator so that only the initial value, $\theta_o = \alpha$, and the constant $\Delta\theta$ be input to obtain not only the first point $(\theta_o, f(\theta_o))$ but also the subsequent points $(\theta_i, f(\theta_i))$, $i = 1, 2, \ldots$. These points would be obtained automatically and in sequence by merely pressing one or two keys successively. The graph of $r = f(\theta)$ is then easily and quickly sketched.

Example 2.
To obtain the graph of the rose $r = 2 \sin 3\theta$ we program this function into the calculator under label [D]. We also program the calculator to give us a sequence $\{\theta_i\}$, with $\theta_{i+1} = \theta_i + \Delta\theta$, where $\Delta\theta$ is arbitrary. The [A] key will give us values of θ_i while the [D] key will give us $r_i = 2 \sin 3\theta_i$. If $\Delta\theta = \pi/12 \doteq 0.26$ and we begin with $\theta_o = 0$, then successively depressing the [A] and [D] keys gives:

θ_i	0	$\pi/12$	$\pi/6$	$\pi/4$	$\pi/3$	$5\pi/12$	$\pi/2$	$7\pi/12$	$2\pi/3$	
	0	0.2	0.5	0.8	1.1	1.3	1.6	1.8	2.1	...
r_i	0	1.4	2.0	1.4	0.0	-1.4	-2.0	-1.4	0.0	...

From this the polar graph of a rose is easily drawn. (Note that here it was assumed that θ is in radians in the function definition.)

3. Limits of Functions

The study of limits of functions is of paramount importance in calculus. Students need to develop a feeling for the meaning of such statements as $\lim_{x \to a} f(x) = L$. Here again the programmable calculator can be of tremendous help to the instructor in illustrating the concept. If we are concerned with the limit of a function, f, as x approaches a finite number, say $x = a$, then, once the function has been defined on the calculator, it is possible to dramatically show the existence (or non-existence) of this limit by choosing elements of the domain of the function sufficiently close but not equal to $x = a$. Here again, the calculator can be programmed to automatically evaluate the function f at any number of values closer and closer to $x = a$. We might begin by evaluating the function at some point x_0 and observe the value $f(x_0)$. Then by successively pushing a RUN button on the calculator, we can obtain the values $f(x_1)$, $f(x_2)$, $f(x_3)$, . . ., where $x_1, x_2, x_3, \ldots$ is a sequence converging to $x = a$. The sequence x_i could be automatically generated by a subprogram in the calculator. A similar procedure can be used to illustrate limits where x approaches infinity. Of course the distinction between one-sided and two-sided limits must be emphasized in these presentations.

On a more sophisticated level, one can even investigate such basic ideas as the $\varepsilon - \delta$ definition of the limit of a function. For example, it is possible to devise a program for the calculator which will attempt to find a value near $x = a$ such that $|f(x) - L| < \varepsilon$ for any given $\varepsilon > 0$. In this case, besides defining the function f in the calculator, the user must supply a proposed limit L, some $\varepsilon > 0$, and an initial value x_0 close to $x = a$. The calculator will then automatically evaluate $|f(x) - L|$ for values of x successively closer and closer to $x = a$, beginning with x_0 and terminating with some x_δ such that $|f(x_\delta) - L| < \varepsilon$. Caution must be used here to assure that f be a monotone function near $x = a$.

Example 3.

(A) Consider $\lim_{x \to -4} \dfrac{2x^2 - 2x - 40}{3x^2 + 15x + 12} = \lim_{x \to -4} B(x)$

where B(x) is the function defined as in Example 1. If B(x) has been programmed on the calculator then the following output sheds light on the limit.

x	B(x)	x	B(x)
-4	Undefined	-3.5	2.27
-4.5	1.81	-3.8	2.10
-4.2	1.92	-3.9	2.05
-4.1	1.96	-3.95	2.02
-4.05	1.98	-3.99	2.004
-4.01	1.996	-3.999	2.0004
-4.001	1.9996		

The limit appears to be 2, which is correct.

(B) If we wish to determine $\lim_{x \to \infty} B(x)$, we can again use the programmed function B on the calculator and evaluate it at several large values of x. The output might be tabulated as follows:

x	B(x)
10	.3030
100	.6270
1000	.6627
10000	.6663
10^5	.6666

The limit appears to be $.6\overline{666}$ or 2/3, which is correct.

Example 4.

Consider $\lim_{x \to 2} \dfrac{x^2 + x - 6}{x - 2}$

Let us use a program which will automatically generate a sequence $\{x_i\}$ converging to 2, beginning at an arbitrary initial x_0.

The function $f(x) = \dfrac{x^2 + x - 6}{x - 2}$ will be successively evaluated at each x_i of this sequence until we decide we have seen enough. In this case, we first define f(x) on the computer. Depending on the input, the results are as follows:

Input: $a = 2$, $x_0 = 1.5$		Input: $a = 2$, $x_0 = 2.5$	
Output: x_i	$f(x_i)$	Output: x_i	$f(x_i)$
1.500	4.5000	2.500	5.5000
1.750	4.7500	2.250	5.2500
1.875	4.8750	2.125	5.1250
1.938	4.9375	2.063	5.0625
1.969	4.9687	2.031	5.0313
1.984	4.9844	2.016	5.0156
1.992	4.9922	2.008	5.0078

The limit appears to be 5, which is easily verified. In this particular limit program, the output of the x_i's can be suppressed by setting a flag. Then only the numbers $f(x_i)$ are obtained successively.

4. Derivatives of Functions

Closely related to the concept of a limit of a function is the idea of the derivative of a function. Indeed, the derivative is in fact a special limit, the limit of the difference quotient

$$\frac{f(x + h) - f(x)}{h} \quad \text{as h approaches 0.}$$

In studying derivatives, one usually begins with a consideration of the slopes of secant lines which go through the two points $(x, f(x))$ and $(x + h, f(x + h))$ on the graph of the function $y = f(x)$. The idea is to begin with h fairly large and then to select successively smaller values of h so that the two points on the curve get closer and closer to each other. The object is of course to define the slope of the tangent line as the limit of the slopes of the secant lines as

h approaches zero. The properly programmed calculator can provide the instructor with values of this difference quotient for any function f at an arbitrary point x and for any desired value h. It can also simulate the limit as h approaches zero, thus giving an approximation for the derivative, f'(x). It should be reiterated here that merely obtaining a good numerical approximation to f'(x) is not the primary objective of the instructor. Of primary concern is teaching the concept of the derivative as a limit of a difference quotient. Hence the calculator should be programmed to provide values of the difference quotient for arbitrary values of h, both positive and negative, or perhaps for a sequence h_i which converges to zero and in this case the intermediate results are as important as the limit.

Example 5.
Consider the difference quotient $\frac{f(x+h) - f(x)}{h}$ for various values of h, and its limit as h approaches zero. We can use a special program where f(x) may be any arbitrary polynomial of degree less than 7. Simply read into the memory registers (R_i) the non-zero coefficients of the polynomial and a value for h. Then for any x, the program computes the difference quotient.

For example, let $f(x) = 3x^4 - 2x + 1$. Store the coefficients 1 in R_1, -2 in R_3, 3 in R_5. Store any $h > 0$ in R_9 and set a flag. Then the following values are immediately obtained ($f^+(x)$ denotes the value of the difference quotient with $h > 0$, $f^-(x)$ denotes its value using $-h < 0$):

x	h	$f^+(x)$	$f^-(x)$	f'(x)
1	.5	19.38	2.63	
	.1	9.72	6.52	
	.01	8.16	7.84	
	.001	8.02	7.98	8.000
-2	.5	-58.6	-129.38	
	.1	-81.24	-95.24	
	.01	-87.30	-88.70	
	.001	-87.93	-88.07	-88

In this program f(x) and f'(x) can also be obtained directly if so desired.

5. The Definite Integral

The study of integration in calculus is usually introduced by considering Riemann sums. To approximate the value of $\int_a^b f(x)\,dx$ we might consider $\sum_{k=1}^{n} f(x_k)\,\Delta x$ where $\Delta x = \frac{b-a}{n}$ and $x_k = a + k\,\Delta x$. It is usually instructive to calculate these approximating sums for various values of n but it can be very time consuming. The programmable calculator can expedite this process and help to demonstrate that the approximation improves as n approaches infinity. It is best in this case to program the calculator so that intermediate results in formulating the Riemann sum be available. In particular, for a given function f which should be defined on the calculator at the start, the input into the program would be the end-points of the interval (a, b) and the value n. Intermediate results which are instructive are Δx, and the values x_k and $f(x_k)$ for $k = 1, 2, \ldots, n$. The final sum $\sum_{k=1}^{n} f(x_k)\,\Delta x$ should of course also be obtained. It is profitable to allow the user the option to suppress the intermediate results. Furthermore, one should provide as an alternative for comparison purposes the computation of the sum $\sum_{k=0}^{n-1} f(x_k)\,\Delta x$ where the function is evaluated at the left endpoints of each subinternal (x_k, x_{k+1}) rather than at the right.

Illustration of the trapezoidal and Simpson's rules for approximating the definite integral are accomplished on the programmable calculator in similar fashion.

Example 6.
To obtain an approximation for $\int_1^5 \frac{1}{x}\,dx$, we may use either of the following two approximating sums:

$$S_0 = \sum_{k=0}^{n-1} f(x_k)\,\Delta x$$

or $$S_1 = \sum_{k=1}^{n} f(x_k)\,\Delta x$$

where $f(x) = \frac{1}{x}$ and $\Delta x = \frac{5-1}{n}$ and $x_k = 1 + k\,\Delta x$

This can be illustrated with a program which will give the following results, depending on which flags are set and what parameters are used. Assume f(x) has been defined on the calculator. Then:

(a) Input: A = 1, B = 5, n = 4
Output with Flag 1 off, Flag 2 on:

$\Delta x = 1$, $x_0 = 1$ $f(x_0) = 1.00$
$x_1 = 2$ $f(x_1) = .50$
$x_2 = 3$ $f(x_2) = .33$
$x_3 = 4$ $f(x_3) = .25$

$$S_0 = \sum_{k=0}^{3} f(x_k)\,\Delta x = 2.0833$$

Output with Flag 1 on, Flag 2 on:

$\Delta x = 1$, $x_1 = 2$ $f(x_1) = .50$
$x_2 = 3$ $f(x_2) = .33$
$x_3 = 4$ $f(x_3) = .25$
$x_4 = 5$ $f(x_4) = .20$

$$S_1 = \sum_{k=1}^{4} f(x_k)\,\Delta x = 1.2833$$

(R/S) 2.0833 (previous value obtained)
(R/S) 1.6833 (average value: S_0 and S_1)

This average value is also the value obtained using the Trapezoidal Rule with n = 4.

(b) Input: A = 1, B = 5, n = 20
Output with Flag 1 off, Flag 2 off:

$$S_1 = \sum_{k=0}^{19} f(x_k)\,\Delta x = 1.6926$$

(with no intermediate results)

Output with Flag 1 on, Flag 2 off:

$$S_1 = \sum_{k=1}^{20} f(x_k)\,\Delta x = 1.5326$$

(with no intermediate results)

(R/S) 1.6926 (previous value)
(R/S) 1.6126 (average value)

The exact value of the integral is

$\int_1^5 \frac{1}{x}\,dx = \ln 5 \doteq 1.6094$, so our results are reasonably close.

Example 7.
An approximating sum for $\int_0^1 \frac{1}{1+x^2}\,dx$ can be obtained as follows, using the same program as in Example 6.

First define $f(x) = \frac{1}{1+x^2}$ on the calculator.

Then:
(a) Input: A = 0, B = 1, n = 5
Output with Flag 1 off, Flag 2 on:

$\Delta x = .2$, $x_0 = .00$ $f(x_0) = 1.00$
$x_1 = .20$ $f(x_1) = .96$
$x_2 = .40$ $f(x_2) = .86$
$x_3 = .60$ $f(x_3) = .74$
$x_4 = .80$ $f(x_4) = .61$

$S_0 = 0.8337$

Output with Flag 1 on, Flag 2 off:

$S_1 = 0.7337$ (no intermediate results)

(R/S) 0.8337 (previous result)
(R/S) 0.7837 (average)

(b) Input: A = 0, B = 1, n = 20
Output with Flag 1 on, Flag 2 off:

$S_1 = 0.7728$

Output with Flag 1 off, Flag 2 off:

$S_0 = 0.7978$

(R/S) 0.7728 (previous result)
(R/S) 0.7853 (average)

Note that the exact value of the integral is $\frac{\pi}{4} = 0.7854$, which confirms our approximations.

6. Sequences and Series

Another important topic in calculus is the study of sequences. Since number sequences $\{u_n\}$ are just functions from the natural numbers into the real numbers, we can treat their limits (as n approaches infinity) just like we treated such limits of general functions. Consequently, if we program the n^{th} term, u_n, of the sequence on the calculator as a function of n, then we can easily investigate $\lim_{n\to\infty} \{u_n\}$. Sequences of functions $\{u_n(x)\}$ can be treated similarly except that they must allow for the input of arbitrary values of the variable x.

Infinite series are of course directly related to sequences. Their convergence and divergence is determined by the behavior of the sequence of partial sums. The programmable calculator can determine the partial sums very quickly and demonstrate convergence or divergence quite effectively. Again, all that is required is a relatively simple program which will display the partial sums S_1, S_2, S_3, . . . iteratively, provided the n^{th} term of the series u_n has been programmed into the calculator. Alternately, the program can be so written that the n^{th} partial sum, $S_n = \sum_{k=1}^{n} u_k$, can be obtained directly without first stopping to display S_1, S_2, . . . , S_{n-1}.

Even more interesting are series of functions $\sum_{k=1}^{\infty} f_k(x)$. These include power series (e.g. Maclaurin and Taylor series). The radius of convergence as well as the rate of convergence of such series is easily demonstrated with the aid of a programmable calculator. All that is required is a calculator definition of $f_k(x)$, the k^{th} term of the series. If the series represents a known function f(x), then it is easy to compare the value of the n^{th} partial sum

$$S_n = \sum_{k=1}^{n} f_k(x)$$

with the correct value of the function for any specified value of x. In general the rate of convergence will depend on the choice of x and indeed the series may converge for some values of x and diverge for others.

Example 8.
Consider the Maclaurin series for the function $f(x) = \frac{1}{(1-x)^2}$. The series is $\sum_{k=1}^{\infty} kx^{k-1}$.

We can use a program on the calculator to obtain the n^{th} partial sum S_n for different values of x and n and compare the result with f(x). First we define the k^{th} term of the series as well as the function f(x) on the calculator. The results obtained on the calculator are tabulated below:

Input:		Output:	
x	n	S_n	f(x)
1	3	6	Undefined
1	5	15	"
1	10	55	"
2	3	17	1
2	5	129	1
2	10	9217	1
0.5	3	2.75	4
0.5	5	3.56	4
0.5	10	3.98	4
0.2	3	1.520	1.563
0.2	5	1.560	1.563
0.2	10	1.562	1.563

It is easily seen that the series fails to converge for x = 1 or x = 2 and that convergence at 0.2 is much faster than at 0.5. (It can be shown analytically that the series converges for $|x| < 1$.)

7. Miscellaneous Application

The programmable calculator can be useful in illustrating almost any iterative procedure. Newton's method for finding roots of a function is a typical example. It should only be necessary for a user to define the function y = f(x) on the calculator. The derivative which is required in obtaining the successive iterations can be computed as part of the main program which displays $x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$ where the initial value, x_0, must be supplied by the user. A numerical solution to the differential equation y' = F(x,y) using the increment method is another recursive scheme which is easily demonstrated with the help of a programmable calculator. Here the function F which must be defined by the user is dependent on the two independent variables x and y. The main program would take any user-defined function and would construct an approximate solution curve y = h(x) beginning at (x_0, y_0), using the formula $y_{i+1} = y_i + F(x_i, y_i)\,\Delta x$. Some initial point (x_0, y_0) must be supplied and x should be arbitrary in the program. Examples for these applications are easily produced but have been omitted here due to space limitations.

Many other applications are of course possible. The interested calculus instructor will no doubt want to devise his own illustrations and perhaps prepare some of his own programs to use with his classes.

Conclusion

Miniaturization in electronics has provided mathematics educators with new tools for the classroom. Programmable calculators which may properly be called micro-computers are now readily available and should not be overlooked in our endeavor to facilitate learning. In this paper we have tried to show that the teaching of certain concepts in calculus can be significantly aided with the help of carefully planned calculator programs. No doubt the capabilities of these micro-computers will continue to increase and this should enable us to make even more sophisticated use of them in mathematics instruction.

REFERENCES

Bell, F. H. Computer-augmented calculus: Its effect upon performance. The University of Pittsburgh, Pittsburgh, Pa. Summer, 1972. Preliminary Research Report.

Bitter, G. G. Computer-oriented calculus. Journal of Educational Data Processing, 1970, 7, 193-195.

Fiedler, L. A. A comparison of achievement resulting from learning mathematics concepts by computer programming versus class assignments approach. Unpublished doctoral dissertation. University of Oklahoma, 1969.

Holoien, M. O. Calculus and computing: A comparative study of the effectiveness of computer programming as an aid in learnig selected concepts in first-year calculus. Unpublished Doctoral Dissertation, University of Minnesota, 1970.

Lang, M. T. Computer extended instruction in introductory calculus. Unpublished Doctoral Dissertation, University of Texas at Austin, 1973.

COMPUTERS IN MATHEMATICAL MODELING

William S. Dorn
University of Denver, Denver, Colorado 80210

ABSTRACT: The impact of computing on mathematics, and in particular on mathematical modeling, has been more profound than most outside observers might expect. The computer not only allows mathematicians to more readily test complex models, but in addition has become an essential ingredient in the actual creation of new and improved mathematical models. Thus the computer has become, in a real sense, part of the creative process of modeling itself, and it is here that the computer has had and continues to have its greatest effect. In this paper we illustrate the use of computers in mathematical modeling through a simple example from population growth.

1. INTRODUCTION

Computers have had and continue to have a profound and far reaching impact both on mathematics and on mathematics education. That impact is nowhere more dramatically felt than in the construction and analysis of mathematical models and in the increasing use of these models across the curriculum. The first thing that comes to mind in this connection is that no longer need we search for models whose solution can be represented by a simple formula. Rather we often turn to more complex and realistic models and content ourselves with tables of computer-generated numbers that in some sense represent the solution of the model.

While not wishing to detract from such 'number crunching' approaches to modeling, we point out that this is one of the least significant uses of computers in modeling. It is true, of course, that such numerical output often leads to the 'discovery' of a formula that can then be verified through mathematical analysis (see e.g. Section 5(a) of this paper). Hence computer experiments may and often do lead indirectly to a better understanding of the particular model or process under investigation.

But the computer's impact on modeling goes much deeper. For we can now test models under much more severe and exacting conditions than ever before. Such stringent tests oftentimes point out weaknesses, if not out-and-out errors, in the model. That in itself leads to greater understanding of the model. But even more importantly, a skillful analysis of these weaknesses may lead to a new and improved model. In some cases it may lead to the creation of an entirely different model. That is to say, the computer can and often does become a part of the creative modeling process itself.

To clarify this argument we turn to a specific example drawn from population growth. Later we will indicate briefly some extensions of these ideas and procedures to other application areas.

2. A POPULATION MODEL

Consider then a single species of life in a closed environment, e.g., a culture of bacteria in a test tube. We set out to model the growth of this population.

The population is to be measured at various distinct points in time, say every day at noon for bacteria. Each such point in time denotes the end of a *period*. We let N_k be the number of individuals alive at the end of the kth period (for bacteria this is noon on the kth day). The change in the population during the (k+1)st period is

$$N_{k+1} - N_k$$

where, of course, k can take on any one of the values 0, 1, 2, 3, ...

We now make an important and crucial assumption. We assume that the number of births (and deaths) during any period is proportional to the number of individuals alive at the start of that period. That is to say, the change in population during the (k+1)st period is

$$AN_{k+1}$$

where A is the proportionality factor and represents the birth rate less the death rate. Generally A is called the *growth rate* (net change in population).

It follows that

$$N_{k+1} - N_k = AN_k \tag{1}$$

and this becomes our mathematical model. The simple BASIC program at the top of the following page produces a list of populations from but two inputs: the starting population, N_0, and the growth rate, A. From the results (not shown) of this program it is readily inferred that if $A > 0$ then the population grows exponentially

```
100 TYPE VALUE FØR A"
200 INPUT A
300 PRINT "TYPE INITIAL PØPULATIØN"
400 INPUT N
500 PRINT "TYPE NØ. ØF PREDICTIØNS"
600 INPUT M
700 PRINT
800 PRINT "PERIØD", "PØPULATIØN"
900 FØR I = 0 TØ M
1000   PRINT I,N
1100   LET N = (1+A)*N
1200 NEXT I
1300 END
```

without bound.

This conclusion regarding rapid unbounded growth can, of course, be verified using elementary results from the theory of linear, first order difference equations (see e.g. Section 2.4 of [1] or Section 3.3 of [8]). But in any case it leads to a rather clear understanding of the model and, at the same time, points out a glaring weakness of the model. If the environment is closed, as we stated it was, the model must eventually breakdown because the limited space will prevent the population from continuing to grow. After all, eventually the test tube will be completely full, and there will be nowhere to place any new bacteria. So our experiment leads us not only to look for a new model but gives us a clue about where to look. In particular we note that it is for large populations (when the test tube is nearly full) that we need to make a correction.

Suppose then that we assume that the growth rate is not a constant but decreases with increasing population. For example suppose we replace the growth rate A with $A - BN_k$ where $A > 0$ and $B > 0$. Then if the population N_k is small, the growth rate is approximately equal to A leading to exponential growth. But if N_k becomes large then the growth rate drops. Indeed for N_k very large, the growth rate could become negative. Equation (1) is replaced by

$$N_{k+1} - N_k = (A - BN_k)N_k \tag{2}$$

A variation of the program at the top of this page (this new program is not shown) can be used to analyse this model. Some typical results are:

```
TYPE VALUE FØR A
?0.5
TYPE VALUE FØR B
?0.0001
TYPE INITIAL PØPULATIØN
?1000
TYPE NØ. ØF PREDICTIØNS
?30

PERIØD          PØPULATIØN
 0               1000
 1               1400
 2               1904.
 3               2493.48
 4               3118.47
 5               3705.22
 6               4184.97
 7               4526.06
 8               4740.57
 9               4863.55
 10              4929.91
 11              4964.47
 12              4982.11
 13              4991.02
 14              4995.5
 15              4997.75
 16              4998.87
 17              4999.44
 18              4999.72
 19              4999.86
 20              4999.93
 21              4999.96
 22              4999.98
 23              4999.99
 24              5000.
 25              5000.
 26              5000.
 27              5000.
 28              5000.
 29              5000.
 30              5000.
```

and

```
TYPE VALUE FØR A
?1.5
TYPE VALUE FØR B
?0.0001
TYPE INITIAL PØPULATIØN
?1000
TYPE NØ. ØF PREDICTIØNS
?25

PERIØD          PØPULATIØN
 0               1000
 1               2400
 2               5424.
 3               10618.
 4               15270.8
 5               14857.3
 6               15069.3
 7               14964.9
 8               15017.4
 9               14991.2
 10              15004.4
 11              14997.8
 12              15001.1
 13              14999.5
 14              15000.3
 15              14999.9
 16              15000.1
 17              15000.
 18              15000.
 19              15000.
 20              15000.
 21              15000.
 22              15000
 23              15000
 24              15000
 25              15000
```

From these results we conclude that apparently the population eventually settles down to some 'equilibrium' value and then stabilizes there (see Section 11.3 of [5] or Section 5.1 of [8]). But suppose we try a very large value for A, say A = 4.

```
TYPE VALUE FØR A
?4.0
TYPE VALUE FØR B
?0.0001
TYPE INITIAL PØPULATIØN
?1000
TYPE NØ. ØF PREDICTIØNS
?8

PERIØD          PØPULATIØN
 0               1000
 1               4000
 2               22099
 3               61658.4
 4              -71884.
 5              -876151.
 6              -8.114486E+07
 7              -6.588532E+11
 8              -4.340275E+19
```

The negative values for the population indicate that the species has become extinct. Yet another flaw has appeared in our model. Or has it?

3. AN ANALYSIS OF THE TROUBLE

Some further experiments indicate that if

(3) $$0 < A < 2$$

then a nice, stable behavior similar to that exhibited in the first two cases above obtains. But then we try the following

```
TYPE VALUE FØR A
?0.5
TYPE VALUE FØR B
?0.0001
TYPE INITIAL PØPULATIØN
?15000
TYPE NØ. ØF PREDICTIØNS
?10

PERIØD          PØPULATIØN
 0               15000
 1               0
 2               0
 3               0
 4               0
 5               0
 6               0
 7               0
 8               0
 9               0
 10              0
```

Here A < 2, but the population has vanished.

The only resolution of this dilemma lies in some relatively straightforward but non-trivial stability analysis (see Section 5.3 of [8]). In particular, one computes the equilibrium population (which turns out to be A/B); lets the population N_0 be equal to the equilibrium population plus ε; linearizes the difference equation in ε; and solves the resulting equation to see under what conditions the population returns to equilibrium. This leads to (3) above. An analysis of the linearization process leads to the condition

(4) $$|\varepsilon| < \frac{|1-A|}{B}$$

The last numerical example given above violates this condition, and this accounts for the instability in that example.

The final result is that: (a) as long as (3) holds; and (b) as long as we do not stray too far from equilibrium (and (4) tells us how far we can safely stray), then the population always returns to equilibrium and is stable.

But even this is not the end of the story. Consider the following example where

$$\varepsilon = 12500 - (A/B) = 12500 - 5000 = 7500$$

This violates (4), yet the population does return to equilibrium and is stable.

```
TYPE VALUE FØR A
?0.5
TYPE VALUE FØR B
?0.0001
TYPE INITIAL PØPULATIØN
?12500
TYPE NØ. ØF PREDICTIØNS
?25

PERIØD          PØPULATIØN
 0               12500
 1               3125
 2               3710.94
 3               4189.3
 4               4525.93
 5               4742.27
 6               4864.49
 7               4930.41
 8               4964.72
 9               4982.24
 10              4991.09
 11              4995.54
 12              4997.77
 13              4998.88
 14              4999.44
 15              4999.72
 16              4999.86
 17              4999.93
 18              4999.96
 19              4999.98
 20              4999.99
 21              5000.
 22              5000.
 23              5000.
 24              5000.
 25              5000.
```

We leave the resolution of this question open for the reader

For completeness we note that for

$$A > 2$$

the solution oscillates above and below its equilibrium value. The larger A becomes, the more violent and wild the oscillations become. For example, the particular case examined at the close of Section 2 above (A = 4) produced oscillations of such force that the population plunged through zero and the species became extinct. Had we used A = 3, the oscillations would have brought the population perilously close to zero, but still the species would have survived.

4. THE PROCESS

The procedure outlined above is typical of much of the use of computers in mathematical modeling. First a mathematical analysis leads to a model that is then tested using a computer. This leads to still more mathematical analysis and another program and to another series of experiments. Then back to the analysis. This sequence of mathematics-computing-mathematics-computing is continued until a sufficient understanding of the process is obtained. Often the computing and the mathematics become so entwined that it is difficult to distinguish one from the other.

We note also that the goal is to understand the model and the actual process. In the example used here the population values themselves were unimportant. It was their interpretation that was of concern.

5. OTHER PHENOMENA AND THEIR MODELS

(a) Epidemics

It is interesting to note that if one studies simple epidemics, a mathematical model similar to (2) arises (see e.g. Section 5.5 of [8]). In particular suppose we consider a private boarding school with S students. When the students return from their Christmas holiday, a few of them, I_0, bring with them a new flu virus to which all other students are susceptible. How fast will the flu virus spread to the other students? If I_k is the number of infected students at the end of the kth incubation period for the virus then

$$I_{k+1} = (1 + CS - CI_k)I_k \tag{5}$$

where C is the contact rate and depends upon such things as the relative closeness or isolation of students, health practices at the school, etc. Equation (5) is remarkably similar to (2). In fact if one replaces CS by A, C by B and I by N then (5) is identical to (2). We should expect, therefore, that the qualitative behavior of (5) should be similar to the behavior of (2), and it is.

Actually since the number of infectives, I_k, can never exceed the total number of students, S; the stability analysis leads to a restriction on the value of C. In particular

$$0 < C < \frac{1}{S}$$

Moreover, if one conducts numerical experiments systematically varying C, S and I_0; one can deduce that the number of new infectives per period never gets beyond

$$CS^2/4$$

This value is of concern to health authorities because it determines the maximum amount of medication that will be needed at any one time.

(b) Predator-Prey (see [7] or Chapter 2 of [2] or Section 5.2.1 of the Instructor's Manual to [8])

Suppose there are two species of life, one of which preys upon the other. If the species are pheasants and foxes, we assume that in the absence of foxes (the predator), the pheasants will grow according to (2), i.e.,

$$P_{k+1} = (1 + A - BP_k)P_k$$

where P_k is the number of pheasants. But foxes eat pheasants. Indeed the more foxes there are, the more pheasants that are eaten. Similarly the more pheasants there are, the more that are eaten (they are easier for the foxes to find). So from the above equation we subtract

$$CP_kF_k$$

to obtain

$$P_{k+1} = (1 + A - BP_k)P_k - CP_kF_k \tag{6}$$

The foxes, on the other hand, are assumed to have only one food supply, pheasants. So if the pheasants die, so do the foxes and quickly.

$$F_{k+1} = DF_kP_k \tag{7}$$

One can, of course, analyse these equations in much the same way as the single species population was investigated. That is to say, we can look for equilibrium populations and then examine their stability. In the end we will probably want to modify the above equations.

In any case we can raise questions such as: When is the most opportune time to hunt pheasants (foxes) if we do not wish to disturb the ecological balance? How many pheasants (foxes) can be harvested? And so on.

(c) The National Economy

A simple model of a nation's economy is given by

$$T_{k+1} = (1+B)AT_k - ABT_{k-1} + G$$

where T_k is the total income of the nation in the kth year, A is a measure of consumer spending rates, B is a measure of investment in private

industry, and G is government spending (see e.g. Introduction of [1] or Section 4.1.7 of [4] or Section 1.3 of [8]). In some cases this economy expands. In other cases it collapses. In yet other cases it oscillates. When and why? If the economy is collapsing, can government spending, G, be increased periodically to prevent the collapse?

(d) Racial Integration

If we let W_k be the percentage of whites in a racially mixed organization at the end of the kth year, then one model of the organization is given by

$$W_{k+1} = (1 - E)W_k + pE$$

where E is the exit rate (retirement, resignation, death, etc.) and p is the probability that someone who leaves is replaced by a white person. An analysis of this equation indicates that W_k never drops below p. Of course, more elaborate models can be and have been constructed. (See Section 1.2.1 of the Instructor's Manual to [8].)

BIBLIOGRAPHY

1. Goldberg, Samuel, *Introduction to Difference Equations*, Wiley, 1958 (paperback)
2. Smith, J. Maynard, *Mathematical Ideas in Biology*, Cambridge University Press, 1968
3. Pielou, E. C., *An Introduction to Mathematical Ecology*, Wiley, 1969
4. Mizrahi, Abe and Michael Sullivan, *Finite Mathematics with Applications for Business and Social Sciences*, Wiley, 1973
5. Malkevitch, Joseph and Walter Meyer, *Graphs, Models and Finite Mathematics*, Prentice-Hall, 1974
6. Dorn, William S., "Simulation Versus Models: Which One and When?", *Journal of Research in Science Teaching*, 12, pp. 371-377 (1975)
7. Dorn, William S., *The Computer as a Mathematics Laboratory*, available from Hewlett-Packard's Educational Division, Cupertino, Calif, 1975
8. Dorn, William S. and Daniel D. McCracken, *Introductory Finite Mathematics with Computing*, Wiley, 1976

A LIBERAL ARTS COURSE BUILT AROUND TIME-SHARING COMPUTER GAMES AND SIMULATIONS

Roy Elliott
California State University at Hayward, Hayward, California 94542

ABSTRACT: How do we give today's college students an understanding of the computer? This paper reports an experiment with one approach: a general education course on a current issue taught using computer games and simulations. The author developed a course, "The Energy Crisis and Computer Games" and offered this course twice during the 1974 - 1975 year. The course proved very popular with students. Experience from the first offering was used to modify the course for the second offering. Results suggest that combining computers with a contemporary issue gives both a sharper analysis of the issue and a deeper feel for the capabilities and limitations of the computer.

TEACHING APPLIED ECONOMICS TO GRADUATE STUDENTS OF BUSINESS AND MANAGEMENT THROUGH THE USE OF A COMPUTER MODEL

Dr. Dale A. Vorderlandwehr
American Graduate School of International Management

ABSTRACT: Graduate students of business and management often confront the economics profession with the criticism that much of the economics they learn cannot be applied. It would be easy to interpret this as another version of the tired litany that economics is too theoretical. However, it seems that the desire is for greater emphasis on the explanation of quantitative economic phenomena in terms of theory. The attempt to satisfy this requirement has lead to the development of the teaching technique dealt with in this paper. That is, the use of a computer macroeconomic model and the success we have had with it as a teaching method. It is used in classes dealing with macroeconomic theory, monetary and fiscal policy, and econometrics.

INTRODUCTION: This paper presents an attempt to teach applied macroeconomics with the use of a computer macroeconomic model. Macroeconomic variables such as GNP, inflation rates, unemployment, etc. are usually studied because of their impact on the microeconomic variables that affect the firm, household, and government agencies. It is desirable, therefore, for students of business and management to gain insight into the workings of macroeconomics, and to forecast macroeconomic variables for government and private planning.

BRIDGING THE GAP BETWEEN THEORY AND APPLICATION: Students of macroeconomics are confronted with economic models that consist of theoretical abstractions. They often encounter difficulty in making a connection between this body of theory and the concrete data such as those of the International Financial Statistics. They will attempt to interpret such data in terms of theory they have learned in a macroeconomics course, but they have trouble seeing the relationships between such data and, for instance, growth theory as expressed in a differential equation. This problem has been part of the rationale behind a concern with the "relevancy" of economics. Because the relationship between theory and specific economic facts is not immediately or easily apparent or mastered, theory has been attacked as irrelevant. It has been argued that the theory is not "applicable" to the realities of economic life. My own experience has been that economic theory is indeed applicable, but the student's criticism may be quite valid in that efforts to teach this aspect of economics have been ineffective.

It is in this particular step of translating an economic model into a definite procedure for interpreting concrete data that the computer macroeconomic model has been effective. Computer related teaching techniques have been in use for some time with varying degrees of success. They are based on the educational principle that the understanding of the relationships between abstract ideas can be strengthened by two related experiences. These are having the student substitute specific examples for the abstract concept, and having the student actively involved in the manipulation of specific examples. The result should be a clarification of cause and effect relationships. The use of such computer models can be very successful if this particular objective is clearly in mind.

Both computer games and computer simulation have been used as teaching methods. The game can be used as a motivational device to capture student interest and involve them with the possibilities of economic decision-making for a country. However, it has the disadvantage of focusing student attention on the objective, of outmaneuvering an opponent, and can easily transform the classroom into a parlor-game environment. It is possible to stress an analytical approach to decisions made in the game, but the game does not lead as directly to analysis as does a computer simulation. The simulation has the advantage of duplicating controlled experimental conditions for a synthetic economic environment approximating a theoretical model. When simulation follows the game, the interest aroused by the game can be maintained to lead into the investigation of analytical relationships of the theory, through manipulation of relevant variables.

THE COMPUTER MACROECONOMIC MODEL:
The macroeconomic model is international in that it is a two-country world model. The comments made in this paper, however, will be restricted to its use in domestic economic analysis. Each country is made up of four economic sectors: Production Sector, Expenditure Sector, Financial Sector and Foreign Sector. The model is standard in that each country's Production Sector has a Cobb-Douglas production function and the flexibility to include a demand for labor hours derived

from business setting the marginal productivity of labor equal to labor's real wage. Labor hours supplied may be exogenously determined or supplied as a direct function of the real wage.

The Expenditure Sector includes real consumption demand as a linear and direct function of real disposable income. Real gross private fixed investment demand may be exogenously given or inversely related to interest and depreciation rates. Government Expenditure is given exogenously and net exports are a function of relative real gross national income and relative prices. Higher income in country one than in country two generates a net export deficit for country one, as will higher prices in country one, and vice versa. Deficits in net exports of each country are subject to production constraints in the net exporting country, however. Real inventory investment is residually determined as the difference between real gross national product (determined in the Production Sector) and the sum of real consumption, real fixed investment, real government expenditure, and real net exports. This is a disequilibrium model in that equilibrium is not determined in the Expenditure Sector in any particular period.

The Financial Sector includes money (foreign and domestic), bonds (government and private, foreign and domestic) and equities in the form of retained earnings. The nominal domestic money supply is given exogenously by each country's central bank, but the real money supply is also influenced by the culmulative balance-of-payments and the price level. The demand for real money balances is directly related to real disposable income and inversely related to the interest rate on bonds.

In addition to real net exports mentioned in the Expenditure Sector, the Foreign Sector also includes international capital flows. Capital flows are of the indirect investment type (domestic residents can purchase foreign bonds). Capital flows are a function of relative interest rates.

THE MODEL AS A COMPUTER GAME:
In courses dealing with macroeconomics theory, policy, and econometrics, the macroeconomics model can be introduced as a Game in the first or second meeting of the class. When using the model in this way, the class is divided into an even number of countries. Each set of two countries becomes one world, and each country includes three students. Each student represents one of three economic agents: labor, business and government. Initially, to add interest to the decisions made by each of the agents, objectives are established for each agent, and each agent is encouraged to outperform the corresponding agent in the foreign country. Typical objectives for the different agents are maximization of real net worth for the business agent, maximization of the sum of real consumption plus real savings for the labor agent and maximization of the sum of real GNP for the government agent.

Usually the labor agent's decision variables are money, wages, and labor hours supplied. The business agent's decision variables are price level, labor hours employed, and fixed real investment. The government agent's instruments are personal income tax rates, profit tax rates, domestic nominal money supply, and real government expenditure.

It is made clear at the outset that the emphasis in the course will be on the analysis of the economic consequences of the decisions made by the agents rather than on the rationale for those decisions. The game approach is used to stimulate interest and to get the student immersed in economic data, but the game is merely the first step toward the use of the computer to simulate economic abstractions. Even the game setting, though, can be used indirectly to examine, for instance, the consequences of economic policies. An example is the game's use to analyze the economic impact of the Kennedy Administration's attempt at keeping real wage percentage increases equal to labor productivity percentage increases.

After a number of periods of play, the students are shown in mathematical terms why the Kennedy proposal would leave labor's relative share of income unchanged. Then the students are asked to calculate the relative rate changes in the real wages, labor productivity and labor's percentage claim on income. The students are able to demonstrate to themselves quantitatively that a relative rate change in the real wage greater than the relative rate change in labor productivity will redistribute income toward the labor sector, that a percentage increase in the real wage less then the percentage increase in labor productivity will redistribute income away from the labor sector and that a situation where the relative rate changes in real wages and labor productivity are equal will leave labor's relative share of income unchaged.

THE MODEL AS A COMPUTER SIMULATION:
Once the student's interest is aroused and familiarity with the economic model is developed through numerous calculations, competitive decision-making, lectures and reading, most of the discretionary decision-making is removed. The game is replaced by a computer simulation, to facilitate student understanding of the relationship between economic theory and economic data.

One area of economic theory that seems particularly difficult for graduate students of business and management (and economics) is growth theory. An example of the use of the simulation in growth theory is drawn from a classic article by Robert Solow, "A Contribution to the Theory of Economic Growth". At one point in the article, Solow attemts to show the impact of constant neutral technological change and a constant rate of increase in labor hours (supplied under competitive conditions in the labor market and a Cobbs-Douglas production function) on output growth, capital stock growth, the capital/labor ratio, real wages, and the relative share of labor's income.

The production function is:

(1) $y=A(t)K^{a}(Loe^{nt})^{1-a}$

where y = real output
$A(t)$ = technological change = e^{gt}
K = capital stock
$L(t)$ = labor hours = Loe^{nt}

The basic differential equation is:

$$\dot{K}=se^{gt}\ K^{a}(Loe^{nt})^{1-a}$$

Where s = saving rate

By solving and examining this differential equation one is able to show that the capital stock increases at the relative rate $n+g/(1-a)$ and that the rate of increase of real output is $n+ag/(1-a)$. It can also be shown that the capital/labor ratio grows at a rate of $g/(1-a)$. And if the capital/labor ratio rises, it follows that the real wage must rise. In spite of a rising real wage, however, labor's share of income is constant at $1-a$.

The computer simulation allows the student to conduct a controlled experiment involving this differential equation. Arbitrary values can be established for the parameter value a, the growth rate of labor n, initial labor hours Lo, initial capital K, saving rate s, and technological change rate g. As the simulation moves through time the student can calculate the marginal productivity of labor and therefore the real wage in a competitive labor market and literally watch the real wage rise through time. Output, capital, capital/labor ratios can all be calculated and compared to the results suggested by the differential equation. Labor's share of income can also be calculated and the constant share conclusion verified.

CONCLUSION:
The author's experience has been that the use of computer games and computer simulation, in judicious combination with other teaching techniques, can increase the student's understanding of abstract macroeconomic theory. These approaches make the relevance of theory more obvious by allowing the student to participate in the process of applying abstract concepts to concret data.

COMPUTER-BASED METHODS FOR LABORATORY INSTRUCTION

J. J. Lagowski
Department of Chemistry
The University of Texas
Austin, Texas 78712

ABSTRACT: A mini-computer has been used to supplement laboratory instruction in general chemistry. The following instructional modes have been employed to create a flexible course which routinely enrolls ~1500 students per academic year: (1) on demand pre-laboratory quizzing (2) interactive evaluation of student results (3) monitoring student progress (4) laboratory experiment simulation. The impact and implications of applying such techniques to an on-going course are discussed, as is a description of typical programs, the strategy of their use, and the results obtained for a two-year period.

INTRODUCTION

A given instructor can be highly successful in teaching his subject if a relatively small number of students are involved. However, as (a) the number of students increases without a concomitant increase in teaching space and qualified faculty, (b) the subject matter becomes more complex, and (c) the instructor is faced with additional responsibilities, it becomes progressively more difficult to make the classical instructional process successful. Often the factors listed above intrude more or less simultaneously in modern science teaching. A very serious problem arises when an attempt to relieve such pressures is made within the constraints of the classical lecture-discussion-laboratory structure commonly used in teaching basic science courses.

Because of the present wide acceptance of providing a higher education for a larger proportion of the population and the improvement of secondary education, colleges and universities are faced with teaching more and more sophisticated concepts to a rising number of students using facilities, both human and technical, that are increasing and improving at a less rapid rate. Typically, the logistics of teaching in such an environment become geared to getting things done easily, and this unfortunately, also leads quickly to a decrease in the attention that can be given to the individual student's needs.

In previous publications (1-5) we addressed the use of computer-based techniques as possible methods of alleviating logistic problems in lecture oriented courses in chemistry. Here we discuss the mediation of computer-based methods in laboratory instruction.

DESCRIPTION OF LABORATORY EXPERIENCE

Although the description of the techniques used to mediate a student's laboratory experience is cast in the light of general chemistry as taught at The University of Texas, it should be apparent that this is only a convenient vehicle for discussion; these techniques can be applied to laboratory instruction in other disciplines.

The general chemistry laboratory at The University of Texas is a separate course, consisting of two, three-hour periods per week for one full semester. Generally 15 experiments are performed which are designed to give the student experience in techniques of physical measurements, synthesis, and qualitative and quantitative analysis: the last named category includes colorimetric, titrimetric and gravimetric analysis. A list of the experiments which the students perform appears in Table 1. Each laboratory section contains 25-30 students under the day-to-day guidance of a laboratory assistant. Typically, 20-30 such sections are taught each semester. It should be obvious that one of the main problems associated with such a course is presenting the students with a uniform and a unified experience.

DESIGN OF A COMPUTER MEDIATED LABORATORY COURSE

There are numerous ways in which computer methods can be applied to the educational process; diverse philosophies motivate and guide their use. We take the position that comp ter techniques are adjunctive and not "main-line", although others are committed to the latter approach. Thus, our approach is to (a) augment, not replace, the existing course, (b) help students learn, not necessarily transfer standard information to them, (c) individualize student experiences, (d) provide the basis for a self-paced instructional approach for the student. These principles are translated into the cycle of student work

shown by the flow diagram in Fig. 1. The flow of work is keyed to the experiments which appear in a commercially available laboratory manual (6).

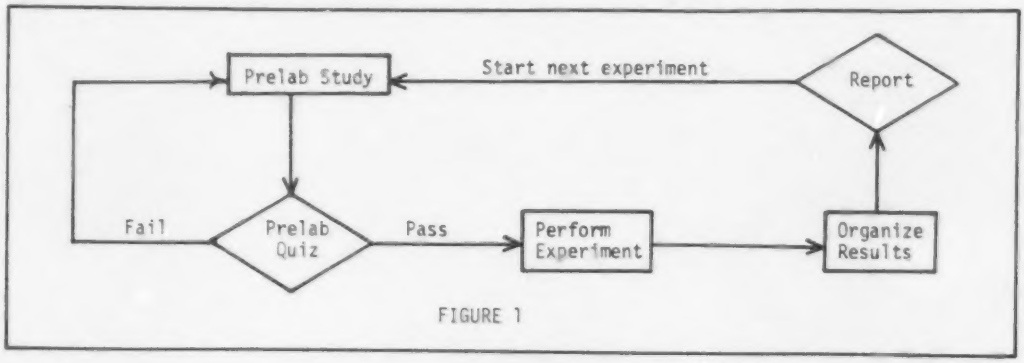


FIGURE 1

Prelab Quizzes. For each experiment the student must pass a prelab quiz designed to establish a minimum level of acceptable performance. These short quizzes (8-10 items) are given on demand on a pass-fail basis (~80% pass level); there is no penalty (except for loss of time) for failure. The questions in the quiz fall into several broad categories, viz., theory, equipment handling, safety, and data manipulation; where possible the questions are generated as opposed to being selected from a list. Fig. 2 shows a typical example of a computer-generated problem; the quantities in the blank spaces in this example are produced by the computer in a random manner.

```
.
.
.
.
C.  Okay, Back to Data Analysis.  I think I'll give you some sample
    calculations.  Suppose a sample of ___ g of KHP required ___ ml
    of NaOH solution to reach the end point.  Knowing that the equi-
    valent weight of KHP is 204.22 g determine the normality of the
    NaOH solution.
S.  .262
C.  Congratulations!  Now using this normality, ___, for NaOH answer
    the next question.  Suppose the titration of a ___ ml aliquot
    of vinegar required ___ ml of standard NaOH.  The normality of
    the acetic acid present is?
S.  .415
.
.
.
.
```

FIGURE 2

When a student passes the quiz the system prints a ticket (Fig. 3) which is presented to the laboratory instructor before the student is permitted to work in the laboratory.

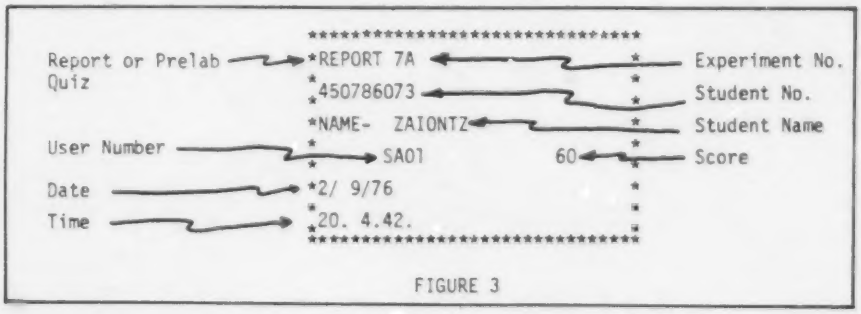


FIGURE 3

Reports. The majority of the experiments have computer reports, i.e., interactive dialogues which discuss each student's results. Nearly all of the experiments have unknowns and the computer system knows the identity of each unknown for each student. The report programs are written in a general way, but when they execute, parameters associated with specific unknowns are set and the dialogue with a student appears to be highly individualized.

Simulation Programs. General chemistry experiments are usually intended to give the student experience in two areas: (a) manipulation, i.e., how to perform certain operations with maximum efficiency so that such operations become a part of the student's repertoire of skills and (b) the collection and analysis of raw data. Computer-based methods are not very useful in the first area, but can be employed with advantage in the second through simulation methods. Computer simulated experiments permit the expansion and compression of time and allow students to collect data on experiments which require complex equipment and expensive or dangerous chemicals. In our simulated experiments we require the student to make all the decisions he normally would be expected to make in the real laboratory (e.g., establishing weights, volumes, time); the student then obtains "raw data" upon which are superimposed all the errors that might be expected if the experiment were really done. Presently we have incorporated three simulated experiments in this course (See Table 1); other are planned.

Experiment 1 (Least Squares Fit) models an electrical circuit in which the student is to determine an unknown resistance (Fig. 4). The student imagines that he passes a current, as measured by the ammeter I,

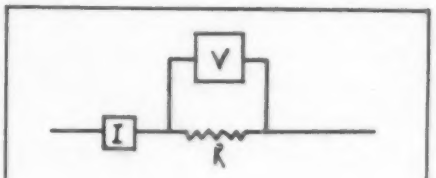


FIGURE 4

through the resistance R and determines the potential drop V across the resistor with a voltmeter which has an unknown calibration error. Thus, the experimental variables are V and I; using Ohm's Law, the value of R can be estimated from the linear relationship called Ohm's Law viz., V=IR. The data obtained by the student, values of V and I, have an experimental error superimposed upon them. In effect the data generator program randomly selects a value of I, calculates the voltage expected for the student's unknown resistance, superimposes error limits on this value, and then randomly chooses a value within these limits. The student is expected to work up his raw data using the conventional least squares approach and report the value of his unknown resistance. The flow of student work is given in Fig. 5. Most students are not sufficiently comfortable with the calculator mode at this point so they follow the upper path indicated in Fig. 5.

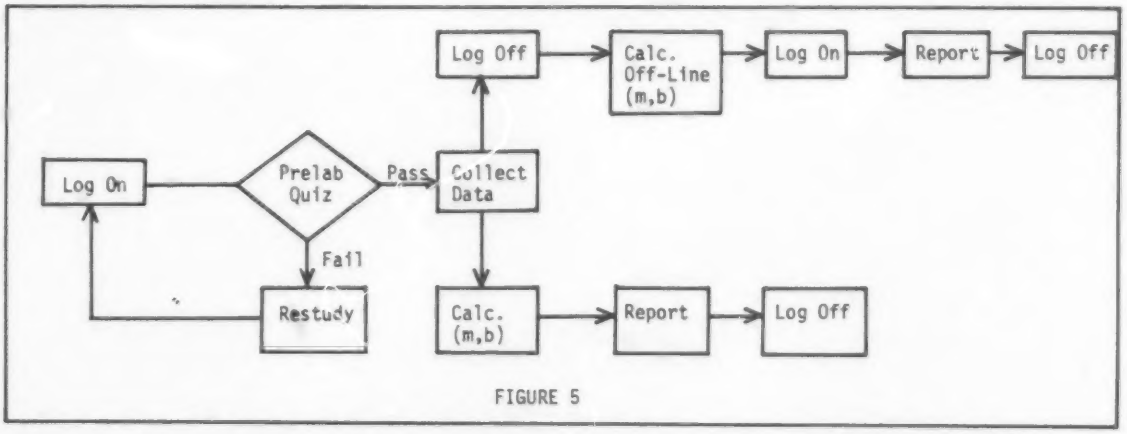


FIGURE 5

Experiment 2 (Identification of a Metal) represents a situation in which the analysis of "raw" data and the conclusions derived therefrom is stressed. The object of the experiment is to identify a metallic element from its physical and chemical properties. The density is obtained from "raw" data obtained using a version of the Archimedes' principle and the student is allowed to test a sample of his unknown

metal with common laboratory reagents, viz., concentrated HCl, HNO_3, and H_2SO_4; dilute HCl, HNO_3, and H_2SO_4; and aqueous NaOH. In addition the student can request quantitative data for any reaction which he discovers to occur with these reagents. The system assigns the unknowns (any one of the 56 metallic elements) randomly and the student is faced with making the decisions shown in Fig. 6.

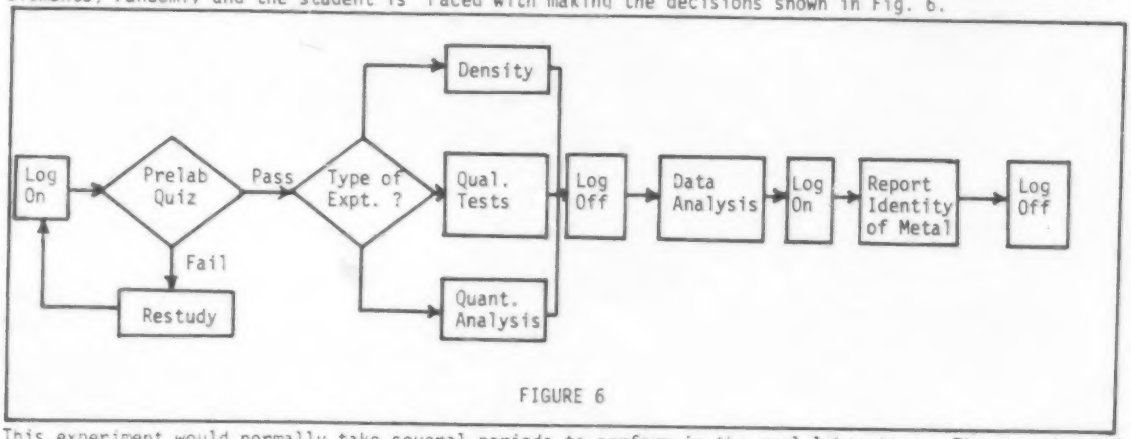


FIGURE 6

This experiment would normally take several periods to perform in the real laboratory. The experimental techniques themselves are trival in that no important manipulative techniques are involved, but they are time consuming. In the simulated version the student still has to make the normal decisions he would be expected to make in the laboratory, but the simulated version is completed in 10-15 minutes. As is the usual case, all "raw" data given to the student has errors superimposed upon them.

Experiment 15 (Observations and Chemical Logic) represents the epitome of usefulness of computer simulated experiments. In this experiment the student is given 8 aqueous solutions containing various substances and 8 labels with the names of the substances. The object of the experiment is to attach the labels to the correct test tubes. The student can only mix the solutions in various combinations, at which time the program gives him the results of his action; no other reagents are permitted. In essence the student is required to establish his own scheme of identification from the observations. Since he knows the identity of the reagents he can predict the observations expected when any combination of reagents is mixed. It then becomes a simple matter to find these observations in his simulated experimental results (if he had the foresight to do them). Of course, there are numerous ambiguous observations (e.g., two or more pairs of reagents that give a white precipitate) which must be resolved at a secondary level.

This experiment has several useful characteristics. (a) Each student has a separate unknown combination which is generated randomly by the program. (b) All students have the same reactions to contend with, thus, there is considerable "cross-talk" among students concerning their unknowns. (c) No one needs to physically prepare the unknown combinations, dispense them, and keep records for grading purposes. (e) No real laboratory time is required. (f) If students do not know what observation to expect for a given combination of reagents, they can either find the results in the literature or do the experiment in the laboratory. In the first instance one could stress the importance of the literature while the second instance involves the experiment more intimately with real laboratory work. Both are valid pedagogical approaches. This experiment is used as an introduction to a real experiment involving the conventional (sulfide-based) qualitative analysis scheme.

Record Keeping. The heart of this course is a record keeping system which automatically mediates the student work in the following ways. A student is not permitted to report on an experiment unless the corresponding prelab quiz is passed. The number of accesses to prelab quizzes are recorded; although no penalty is extracted for multiple accesses, this information is useful in counseling students. A student may report on his experimental results several times, a penalty being imposed each time he does so; this operation is automatic and the penalty can be variable within, as well as across, experiments.

Student Access to the Computer System. Each student in the course is issued a separate user number which is keyed to the record keeping system. When a student logs on he immediately has the choice of working on a variety of programs (Figure 7).

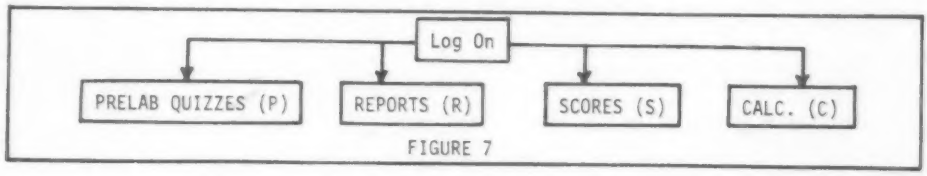


FIGURE 7

The nature of the prelab quizzes (P) and the reports (R) is discussed above: SCORES (S) is the program which shows the student a running inventory of his work. The student can look at (only) his record but he cannot alter any items. The instructor can look at any record and can alter records if the need arises. CALC (C) is the equivalent of a fairly sophisticated hand calculator. Students can call CALC at any time, even within a running program. When he is finished with CALC, the student is automatically returned to the same place in the running program from which he came.

RESULTS

The application of these computer-based techniques to laboratory instruction has resulted in the following benefits. (1) Students are better prepared when they perform their experiments than they were previously. Noticeably less "cook-booking" occurs in the laboratory. (2) Evaluation of results is more even-handed over the large population of students (~900 students in Spring 1975) working under the immediate supervision of a large number of laboratory assistants who possess a wide variety of backgrounds. (3) Students learn to work efficiently at their own pace. (4) Simulation methods have permitted students to have laboratory-type decision making experiences which are normally constrained by time, availability of equipment or chemicals, or safety considerations.

COSTS

Two types of cost must be considered in the implementation of computer-based methods. (1) Development costs incurred in authoring and debugging the programs and (2) production costs which are the recurring costs necessary to keep the programs running in an educational environment. Conceptually, developmental costs are heavy in the area of personnel whereas student use costs are weighted in the area of computer time and communications. Computer costs depend upon the quality and availability of time-sharing in the local environment. Our earlier work had been supported on a large general purpose time-sharing system (CDC 6600/6400). Recently we have been exploring the use of time-sharing minicomputers. Inspection of the data available shows an average development cost of ~$200/program, a number which includes computer, communications, and personnel costs. It is important to note that although very careful planning and design go into the development of a program, its lifetime is sufficiently long to warrant this expense. Following the initial development of a program, the only additional cost comes from its up-dating or revision which is usually negligible unless a major change in strategy occurs. Our experience on student use costs with the large general purpose computer system indicates that interactive educational programs can run on such systems for as little as $0.21 per student hour. Such calculations are, however, highly sensitive to the distribution of uses and the corresponding sources of funds for the programs that are running on the local computer. Charging algorithms, which have a local character by necessity, must reflect the efficiency of the systems and the distribution of income among personnel, rental costs, equipment purchase, etc. Accordingly, cost statements based on computer usage are not useful indicators from one educational environment to the next. Recently, we have used a time-sharing minicomputer system (20 ports) to run education programs. The system is completely dedicated to such use and present data suggest that the student use cost will be approximately $0.50/hour.

Acknowledgments. The work described here was supported by a joint National Science Foundation and The University of Texas Grant (PROJECT C-BE).

REFERENCES

1. S. Castleberry and J. J. Lagowski, "Individualized Instruction Using Computer Techniques," J. Chem. Ed. 47, 91 (1970).

2. S. J. Castleberry, E. J. Montague, and J. J. Lagowski, "Computer-Based Teaching Techniques in General Chemistry," J. Res. Sci. Teach. 7, 197 (1970).

3. L. B. Rodewald, G. H. Culp, and J. J. Lagowski, "The Use of Computers in Organic Chemistry Instruction" J. Chem. Ed., 47, 134 (1970).

4. G. H. Culp and J. J. Lagowski, "Studies Involving the Application of Computer Techniques to Undergraduate Organic Chemistry Instruction," J. Res. Sci. Teach., 8, 357 (1971).

5. S. J. Castleberry, G. H. Culp, and J. J. Lagowski, "The Impact of Computer-Based Instructional Methods in General Chemistry," J. Chem. Ed., 50, 469 (1973).

6. "An Introduction to Chemical Practice," Third Edition, J. J. Lagowski, Ed., Burgess, Minneapolis, 1975.

TABLE 1

EXPERIMENTS PERFORMED IN GENERAL CHEMISTRY LABORATORY

1. Least squares fit*
2. Identification of a metal from its physical and chemical properties*
3. The density of a liquid
4. Determination of the boiling point of a mixture of liquids
5. Synthesis of trans-dichlorobis(ethylenediamine)cobalt(III) ion
6. The temperature dependence of the rate of aquation of trans-dichlorobis-(ethylenediamine)cobalt(III) ion
7. Qualitative analysis of cations
8. Colorimetric determination of copper
9. Identification of elements from atomic emission spectra
10. The determination of the acidity of a solution of vinegar: an introduction to acid-base titrations
11. The determination of iron in a sample: an introduction to redox titrations
12. Semi-quantitative separation of a two-component organic mixture by extraction
13. Colorimetric determination of an equilibrium constant
14. Gravimetric determination of nickel
15. Observations and chemical logic*

* Simulated experiments